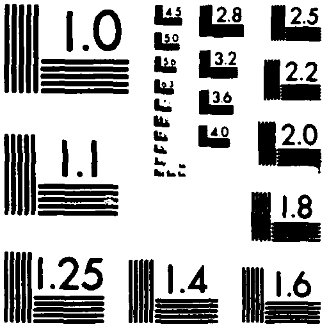
AD-A182 692 A DATABASE DESIGN FOR THE BRAZILIAN AIR FORCE MILITARY 1/2
PERSONNEL CONTROL SYSTEM(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. W MUSSATO
UNCLASSIFIED JUN 87 AFIT/GCS/ENG/87J-1 F/G 5/9 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

AF INSTITUTE OF TECHNOLOGY

A DATABASE DESIGN FOR THE BRAZILIAN AIR FORCE MILITARY PERSONNEL CONTROL SYSTEM

THESIS

WAGNER MUSSATO
Major, BRAZILIAN AIR FORCE

AFIT/GCS/ENG/87J-1

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

1

AFIT/GCS/ENG/87J-1

A DATABASE DESIGN
FOR THE BRAZILIAN AIR FORCE
MILITARY PERSONNEL CONTROL SYSTEM

THESIS

WAGNER MUSSATO
Major, BRAZILIAN AIR FORCE

AFIT/GCS/ENG/87J-1

AFIT/GCS/ENG/87J-1

A DATABASE DESIGN FOR THE BRAZILIAN AIR FORCE MILITARY PERSONNEL CONTROL SYSTEM

THESIS

Presented to Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Computer Systems

Wagner Mussato, B.S.
Major, BRAZILIAN AIR FORCE

June 1987

Approved for public release; distribution unlimited

Preface

The purpose of this thesis was to design and partially implement a database application for the Headquarters of Brazilian Air Force Military Personnel Control System.

This extensive, hard, but rewarding work, could not be finished without a great deal of help from others.

I wish to express my sincere appreciation to my advisor, Capt. Mark Roth, who reviewed this work so many times, for his patience and dedication in making valuable comments and suggestions. I wish also to thanks my thesis committee, Dr. Thomas C. Hartrum, Dr. Henry B. Potoczny and Capt Carl Davis, for their important refinements in this work.

My special thanks for the Brazilian Air Force for allowing me to have the experience of being in touch with such advanced technology that will be used when I return to my country.

I want to dedicate this work to my family, to my loving and often times neglected wife Thais, my son Alexandre and my daughter Tatiana. I want to apologize for those nights and weekends when studying took precedence over you. Please never forget, I love you so much.

Wagner Mussato

Table of Contents

Page

Preface ii

List of Figures v

Abstract vi

I. Introduction 1

1.1 Background 1
1.2 Problem Definition 2
1.3 Summary of Current Knowledge 3
1.4 Scope 3
1.5 Assumptions 4
1.6 Approach 5
1.7 Sequence of Presentation 8

II. System Requirements 9

2.1 Areas of Concentration 10
2.2 General System Requirements 14
2.3 Database Requirements 15
2.4 Data Elements 16

III. Conceptual Design - Entity Relationship Model . 18

3.1 Data Models 18
3.2 Entity Relationship Model 20
3.3 Personnel Headquarters' ERM 22
3.3.1 Identification of Entities . . . 24
3.3.2 Identification of Relationships . 25
3.3.3 Identification of Attributes . . 27
3.4 Personnel Headquarters' ER Diagram 28

IV. Normalization Process 30

4.1 Creating Relations from ER Diagram 30
4.1.1 Entity --> Relation 31
4.1.2 Weak Entity --> Relation 32
4.1.3 Relationship --> Relation 33
4.2 Normal Forms 34
4.3 Functional Dependencies 35
4.4 First, Second, and Third Normal Forms . . . 37
4.4.1 Fist Normal Form (1NF) 37
4.4.2 Second Normal Form (2NF) 40
4.4.3 Third Normal Form (3NF) 42
4.5 Boyce/Codd Normal Form (BCNF) 44
4.6 Fourth Normal Form (4NF) 45

V. Prototype Implementation 49

5.1 Prototype Considerations 49
5.2 Selected Relations 50
5.3 Selected Computational Environment 54
5.3.1 Structured Query Language (SQL) . 55
5.3.2 Common Business Oriented Language (COBOL) 56
5.4 Decision Support System (DSS) 57
5.5 Prototype Storyboarding 59
5.5.1 Main Menu 60
5.5.2 Inclusion / Addition 61
5.5.3 Select 66
5.5.4 Deletion 70
5.5.5 Modification 73
5.6 DSS Queries 75

VI. Conclusion 78

6.1 Conclusion 78
6.2 Recommendations 78

Appendix A: Personnel Database Data Dictionary . . . 83

Appendix B: Description of Entities and Relationships 91

Appendix C: ER Diagram for the Personnel Headquarters 103

Appendix D: Analysis of the Relations Until BCNF . . 116

Appendix E: Prototype Programs 142

Bibliography 170

Vita 173

List of Figures

Figure Page

1. Air Ministry Hierarchy 9

2. Data Flow Diagram 11

3. Relation Course 32

4. Relation Ext_Num 33

5. Relation Per_Pro 34

6. Normal Forms 34

7. Functional Dependencies in the Relation Percourse 41

8. ER Diagram - Prototype 52

9. ER Diagram - Prototype (cont) 53

10. The Connotational View 58

11. Main Menu 60

12. Inclusion / Addition Menu 62

13. Selections Menu 67

14. Deletion Menu 70

15. Modification Menu 73

16. Architecture for End-User Computing 81

17. Example of Integrated Facilities 82

Abstract

This thesis addresses a database design with partial implementation for the Brazilian Air Force Military Personnel Control System. After defining the problem and specifying requirements, the conceptual design was performed using Entity Relationship Model. After defining the Entities and Relationships, the Normalization Theory was used to ensure that all relations met the constraints of the Fourth Normal Form (4NF).

During the implementation phase, a prototype was implemented using Oracle DBMS, with SQL as a query language, and Cobol as a host language. The decision to use this environment for the implementation was made because SQL and Cobol are languages used in the Brazilian Air Force.

Finally, recommendations were proposed for future research in this area, along with an optimal environment for a database user, combining mainframe and personal computers.

[illegible] Systems [illegible]

A DATABASE DESIGN
FOR THE BRAZILIAN AIR FORCE
MILITARY PERSONNEL CONTROL SYSTEM

I. Introduction

1.1 Background

The first system using Electronic Data Processing (EDP) to control military personnel in the Brazilian Air Force was built in 1968. Because it was the first system developed, the requirements were not clear and the system did not address all user needs. This system was not developed in a higher level language, but rather, used assembly language to record, on magnetic tape, all information related to military personnel.

By using this EDP system, military personnel control became more efficient. However, when changes began to be generated by users, the system showed its inflexibility by forcing too much work on programmers attending to users' requests.

To partly solve this problem, another system was developed to complement the existing system. Military personnel headquarters did not change the whole system because the existing one was still working without problems in several areas. Thus, the new system was developed to provide the military personnel headquarters with an important type of printout not supplied by the existing system.

With this new system, the users' requirements were satisfied. However, EDP personnel had an increased workload. Instead of maintaining only one system, they were maintaining two systems; consequently, the number of maintenance personnel increased (29).

In the early 1980's, military personnel headquarters decided to create one unique system to control military personnel, from admission into the Air Force, until retirement. The system was divided into modules. The first module handled the admission process. Since the majority of new personnel go straight to the reserve forces without going on active duty, this new system was developed using the conventional file processing system approach rather than employing database techniques.

The new system was completed in 1983. At the same time, another group was working with users to create a data flow diagram for the processes and data elements necessary to control military personnel on active duty. This data flow diagram has also been completed, and is the starting point for design of the database in this thesis.

1.2 Problem Definition

The problem was to increase the efficiency of military personnel control, by employing one database system to replace the two existing systems. Such a system will provide flexibility, reliability and efficiency not only for the people at operational level, but also for the decision makers of the military personnel headquarters. The objective of this thesis effort was to design the entire database and to implement a prototype of the personnel database for the Brazilian Air Force.

1.3 Summary of Current Knowledge

The current database application systems existing in the market that handle personnel control, do not apply for this research, because this database deals with information only concerned with the military. Current knowledge in controlling military personnel is found in the already existing data flow diagram that will be the starting point in the design phase of the project.

Due to the characteristics of the information to be handled, not many bibliographical sources are available for research; however, information and techniques published on similar database designs will be helpful to support this research project (4;6;14;21;26;27;28).

1.4 Scope

This thesis deals with personnel information generated from military organizations and information generated within the military personnel headquarters.

Personnel information involving money, such as payroll, will not be covered, since this type of information is handled by the military finance headquarters (DIRINT). Therefore, since that organization will in the future make use of the same database, the links for further development will be provided in the military personnel database.

The main effort in this thesis will be dedicated to the design function, with focus on the identification of the entities, attributes, and the relationships among them. Due to the size of the project and time constraints, the whole database will not be implemented, but only selected entities, attributes, and relationships, as representative of the entire database will be implemented.

1.5 Assumptions

In the development of a database system, the starting point is the study of the environment, and documented assumptions for it (3:335). This study was done by a team that looked at the military personnel headquarters and decided on a logical project, in which all manual routines and exist'ng related documentation were analyzed. As a final product of that analyses, a data flow diagram was drawn and that documentation will be the starting point for the design of the database (13).

The relational structure will be used in the implementation phase of the project, because of the following advantages:

Simplicity - A relational data model has the structure very similar to what the user sees, and its physical implementation does not have to be concerned with lower level type of constraints, such as pointers, common in other models.

Data Independence - The relational data model does not have to be concerned with details of storage structure and access strategy, and provides a higher degree of data independence than hierarchical and network models. Therefore, the design of the relations must be more complete than other data models.

Theoretical Foundation - Unlike the hierarchical and network models, the relational data model is based on the well defined theory of relations. There are formal query languages such as relational algebra and relational calculus (27:17-22). By using normalization, the relational model provides a stronger foundation than other models during the design phase (3:94).

The disadvantages of a relational model (3:95) would be related to performance, but, since several techniques of query optimization can be used, the problem can be minimized in terms of software. The technological improvements in building faster hardware could be used as an argument that the disadvantages will soon no longer exist.

1.6 Approach

The first step in designing a database for the military personnel headquarters of the Brazilian Air Force was to analyze the existing Data Flow Diagram for that organization. The objective was to collect all necessary information about the usage, relationships, and meaning of each data element. Thus, a data dictionary was created in order to control and manage the data elements and their respective meanings.

The first four types of information included in the data dictionary were as follows: code, name of the data element in Portuguese, name in English and Description. Afterwards, data type and range were also included (Appendix A).

Since there was no software available at AFIT to handle a database data dictionary, it was not automated.

With all data elements already defined, the next step was to create an Entity Relationship (ER) Model, where the attributes, entities and the relationships among them were identified. In order to create such a model, these relationships were considered:

1 - one-to-one relationship between two entities,

2 - one-to-many relationship between two entities,

3 - many-to-many relationship between two entities.

As a final product of the ER model, an ER diagram was drawn in order to express graphically the model.

The next step towards creating a conceptual model was the normalization process, that is, the process of grouping the data elements into a set of relations (tables), representing attributes, entities and relationships.

During the normalization process, the relations were analyzed to ensure that the conceptual model worked. The analysis was done in order to avoid violations of fourth normal form. Since this thesis project dealt with data used in the real world, some decompositions during the normalization process, that seemed unclear, were provided with the necessary explanation. Due to the large amount of data elements to be analyzed, much of this thesis effort was concentrated in this process.

The next step was the implementation phase and, as stated before, only one portion of the entire database was implemented. For that phase, the relational DBMS ORACLE was used to store the relations and, the "COBOL" language was used to code the application programs to manipulate the DBMS. The hardware used was the HARRIS 800, running Virtual Operating System (VOS).

During the implementation phase, several tools and techniques of software engineering were applied, mainly in the software design. Techniques such as structured programming, stepwise refinement, integrated top-down development and modularization criteria were used in the programming phase (18:137-179).

The objective of this thesis project was not only to address the operational level of the personnel headquarters, but also their decision makers. In this case, some concepts of a decision support system, such as Dialog Design Techniques were applied. Others, such as Interactive Design, could not be applied since the decision makers could not be reached during this thesis work. However, they will be continued as soon as the system is being implemented in the personnel headquarters (30:39,219).

1.7 Sequence of Presentation

Following the introductory chapter, where the problem was identified, Chapter II contains the description of the system and its requirements.

Chapter III, Conceptual Design, presents the Entity Relationship Model, with the identification of the existing entities, weak entities, and relationship among those entities. In this chapter the attributes of the ER Model were also identified. Appendix C presents a complete Entity Relationship Diagram for the entire database.

Chapter IV, Normalization Process, presents the process toward creating a Conceptual Model. In this chapter a complete analysis of the system, using the normalization theory, is done to ensure that the system satisfies the constraints of the normalization process, up to fourth normal form (4NF).

Chapter V, Prototype Implementation, presents the selected relations implemented, explains the reason for selecting the DBMS and language used, and shows the complete set of screens designed for the prototype.

Finally, Chapter VI, Conclusion, presents the conclusion of the overall research, and makes recommendations about the implementation of the prototype and new research to be developed in the area.

II. System Requirements

In this chapter, the military personnel control system is described. As can be seen in Figure 1, Personnel Headquarters function under the hierarchical subordination of the Personnel Majcom, and at the same level of Financial and Health Headquarters.

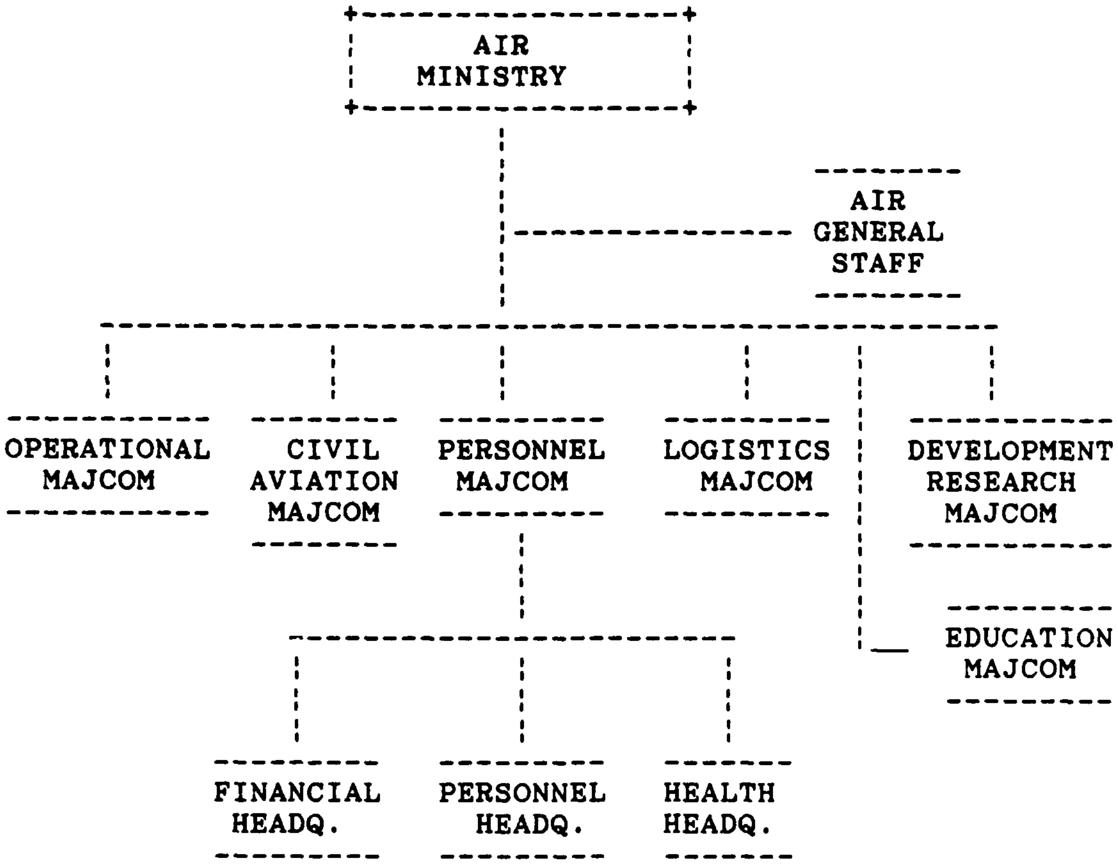


Figure 1 - Air Ministry Hierarchy

As stated in the previous chapter, the scope of this database design is only the military aspect of the Personnel Headquarters. Financial and Health Headquarters information will be used only if it is strictly related to military personnel control.

As the system is being implemented, each one of these headquarters will become part of the database.

Figure 2 presents the complete Data Flow Diagram (DFD), for the personnel headquarters (19). It was left in Portuguese for two reasons, to preserve the original work, and to show that most of the inputs to the personnel control come from outside the personnel headquarters, as can be seen by examining the square boxes outside the large rectangle. Those boxes mean another military organization sends information to the system, or receives information from the system, depending on the arrow direction.

Within the Personnel Headquarters there is also a difference between military and civilian personnel. Because they have different types of control, this database will not address the civilian segment. The same DFD is currently being developed for the civilian personnel to be implemented later on.

2.1 Areas of Concentration

After analyzing the DFD for the military personnel control system, several functions can be seen as areas of concentration of particular information. These areas are:

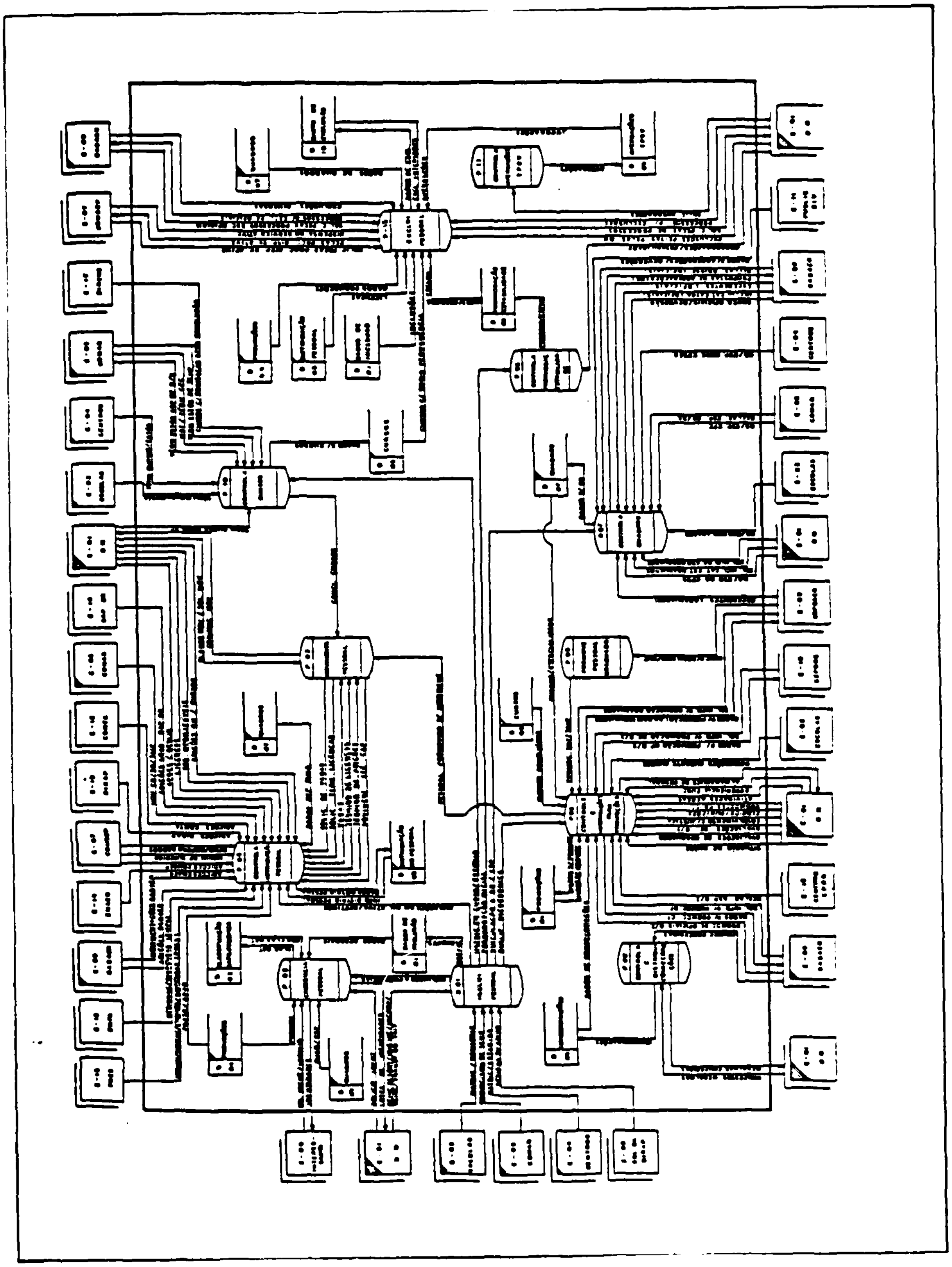


Figure 2 - Data Flow Diagram (DFD) (13)

P01. Personnel Admissions: control the admission of the military personnel in the Brazilian Air Force. Those admissions come from several sources, such as schools of formation, in the case of officers and sergeants, and several other organizations in the case of airmen.

P02. Personnel Records: keep the basic records with information such as that used for the identification card.

P03. Personnel Moving: this area receives all the necessary inputs in order to select the next assignment. This is one of the most active and critical areas, because of the large variety and amount of information to process. For its importance this area was selected to be implemented as a prototype for the entire system.

P04. Control Personnel Distribution: record all information concerned with personnel distribution, that is, where the person is located (organization, squadron). This is one of the most valuable sources of information to the personnel moving (P03) area.

P05. Sergeant Promotion: processes the promotion of a sergeant.

P06. Control Information for Promotion: receives, records and control all information related to promotions of sergeants and officers through the rank of captain.

P07. Control Active List: the Brazilian Air Force controls the military, not only by rank, but also by specialty. The Control Active List, controls personnel according to each specialty, such as, pilots, engineers, medical doctors, etc.

P08. Control Stability: controls the process of stability of the military in the Air Force. Such control is mainly used for sergeants, who have to renew their contracts with Air Force within certain periods of time. After acquiring stability (10 years), no more contracts have to be renewed.

P09. Control and Distribute Medals: controls the process of distribution of medals.

P010. Control Courses: controls the internal and external courses related to the Air Force.

P011. Control Length of Service: controls information related to length of service.

P012. Personnel Exclusion: controls the records and the process of exclusion of the active duty military. Such exclusion can be for retirement, leaving the Force, health problems, etc. Another system (Reserve Forces), will take care of the military after the active duty.

2.2 General System Requirements

In order to develop a new system for the personnel headquarters, it was determined that a general planning of systems should meet the following general system requirements:

1. The new system should be designed to increase the efficiency of the management of information.

2. Since the entire system will not be implemented at once, a top-down approach should be used in the design phase to allow modular implementations.

3. Extensive use of terminals by means of query languages should be used in order to increase the user friendliness as much as possible, without losing efficiency.

4. The new system should be designed in order to avoid redundancy as much as possible.

5. The new system should be designed in order to assist the three levels of information: strategic - top managers, tactical - middle level managers, and operational - common users.

6. The new system should be able to allow sharing of data among their users. In order to avoid duplicity of updating, definition of responsibility must be

provided, to make sure that each data element will be updated by the correct user (29:8).

Such requirements, although not specifically addressed during this thesis project, were general goals for the conceptual design.

2.3 Database Requirements

In order to satisfy the general system requirements, and after some analysis had been done on the DFD, it was determined that the design of a database application should be performed. The database should be designed, to consider the following requirements:

1. It is necessary to store, not only the present information, but also, the historical values of some data, in order to perform some specific queries.

2. On-line data entries will be the primary source of input data to the system. This capability must be considered, including some error detections that must be checked at this point.

3. As a user friendly system, on-line queries must also be supported by the system.

4. Some outputs from the system must be done by means of reports, mainly because of their size and also due to the personnel system requirements specifying

several reports on paper.

5. A high level of security must be addressed by means of a view mechanism, allowing only authorized users to address the data to be updated (15:437-444). Integrity of the database must also be addressed, by defining which users will update, add, and delete the data element.

6. System and media failures must be addressed by using available tools, such as backup and restore. Concurrency problems, such as a deadlock situation, must also be considered, since the database will have multiple users (15:413-433).

2.4 Data Elements

In identifying the data elements, one of the most common problems is the existing redundancy that needs to be avoided (32:45). Since the DFD gives the necessary information to avoid redundancy, this type of problem will not be addressed.

By analyzing the Data Flow Diagram, a list of all data elements used in the entire system was built, and can be found in Appendix A, with the following types of information:

1. Code: four alphanumeric positions, where the first two indicate the area where the data element is

mostly used, and the remaining two positions are numeric from 00 to 99.

2. Name in Portuguese: since this system will be implemented in the Brazilian Air Force, this information will be helpful during the implementation phase in Brazil.

3. Name in English: actual name that each data element will be referred to during the development phase of the system.

4. Description: small description of the content of each data element, in order to avoid doubt about its meaning.

III. Conceptual Design - Entity Relationship Model

Conceptual design deals with information independent of any actual implementation, (i.e., any particular hardware or software system). To develop a database that satisfies today's as well as tomorrow's information needs, a conceptual model must be designed (3:124).

The main purpose of the conceptual design is to represent information in a form that is comprehensive to the user. The conceptual design reflects the data processing needs of the organization, and is represented by entities and their relationships.

3.1 Data Models

One of the major responsibilities of the database designer is to develop a conceptual model of the organization. This model is a communication tool between the various users of data, and should be developed without any concern for physical implementation.

As pointed out by Teichroew (31), the first generation of data models, hierarchical, network, and relational, have all been used as basis for database management system (DBMS), though relational DBMS have only recently become commercially available. Teichroew considered also the second generation of data models that have been proposed in recent years (7).

A second generation of data models has been used because previous models were considered too "low level" for adequate modeling of the real world. A brief description about some of these data models will be explained below.

One of the new data models used is the Semantic Data Model (SDM), developed by McLeod (25), and Hammer and McLeod (20). This model provides a class of real world semantics which are important in data models.

The SDM considers a database to be a collection of entities which may be objects (concrete or abstract), events (point event or duration event) or names which are designators for an object or event.

Entities are organized into classes, which are meaningful collections of relevant objects. Objects have properties. A property is a characteristic of an object. Officer's name and rec_num (record number) are examples of properties of the object officer within entity personnel. The entities and classes have attributes which describe their characteristics, and relate them to the other entities.

The Entity_Link_Key_Attribute (ELKA) modeling technique, is part of a general methodology for constructing a model of an integrated engineering and manufacturing system (5). The ELKA modeling technique uses entities, links and attributes. An entity is an object which is described by properties whose values can be considered as remaining fixed over some period of time. Link is a direct connection between two entities. Attributes are properties of entities. Some of the attributes of an entity may be Key attributes of the entity.

The Structural Model (SM), proposed by Wiederhold and El-Masri (34), is a relational model which uses relations as "building blocks", and includes two extensions. Logical connections are defined between the relations, and relations are classified into relation types (primary, reference, net, and lexicon relation).

In the Entity Relationship Model (ERM), Chen (8,9) proposed to model the real world in terms of entities, relationships, and attributes. The ERM was the selected model used in this database design because it is a worldwide accepted model. Further explanations about this model will follow in the next section.

3.2 Entity Relationship Model (ERM)

Some terms must be defined before considering the details of the ERM. The first term is Enterprise, which is any kind of organization, such as a bank, a university or a personnel headquarters. Entity is a "thing" which can be distinctly identified. Entities can be classified into different types, concrete, such as person (personnel) or place (unit), or abstract, such as course. Relationship is the connection between entities. Personnel_course is the relationship between two entities personnel and course. Attributes are characteristics of each entity or relationship. Name is an attribute of an entity personnel and date_end_course is an attribute of the relationship personnel_course.

Many versions of the ERM have been proposed in the recent years for use in information modeling and analysis. The difference among them are their various interpretations of the concepts of entities, relationship and attributes.

Entity Relationship Models can be divided into two major categories based on the type of relationship allowed in the model (10).

1. Generalized (N-ary) Entity-Relationship Model (GERM), which allows relationships among more than two entities.

2. Binary Entity-Relationship Model (BERM), which allows relationships between only two entities.

Each of these two categories are subdivided into three subcategories, depending on their treatment of attributes:

1. Model allows attributes for both entity and relationship.

2. Model allows attributes only for entities.

3. Model does not allows attributes at all.

For the purpose of this thesis work, GERM was the selected category to be used because it allows relationships between more than two entities. The selected subcategory was the model that allows attributes for both entities and relationships.

3.3 Personnel Headquarters' ERM

By using the data elements identified for the personnel headquarters (Appendix A), the first step was to group the data elements by similarity, i.e., those that seemed to be related. With the data elements grouped, the process of identifying the entities became less difficult.

The list of data elements grouped is as follows:

Personnel : name, rec_num, state_birth, dt_birth, father_name, mother_name, id_num, inc_tax_num, soc_sec_num, med_rec_num, tag_name.

Moving: unit_dest, dt_moving, dt_pres, dt_detach, unit_mov, sit_unit, sum_mov.

Designation: unit_desig, dt_desig, dt_waiver, sit_unit_des, sum_desig.

Nomination: unit_nom, dt_nom, dt_exo, sit_unit_nom, sum_nom.

Attachment: unit_attach, rea_attach, dt_start_attach, dt_end_attach, sit_unit_attach, sum_attach.

Leaving: type_leave, dt_start_leave, dt_end_leave.

Promotion: rank, dt_prom, crit_prom.

Flight: year_ref, qua_ref, hs_diu_1p_qua, hs_diu_2p_qua, hs_diu_of_qua, hs_noc_1p_qua, hs_noc_2p_qua, hs_noc_of_qua, rank_flight, num_ifr_card, unit_ifr, dt_exp_ifr_card, unit_health, dt_exp_health.

Course: type_course, dt_start_course, dt_end_course, grade_course, classif_course, area_course, name_course, level_course, fin_course.

Medal: type_medal, dt_medal, dt-decennium, grade_medal, sum_medal.

Active list (Specialty): active_list, res_non_dut_sta, dt_non_dut_sta, dt_return, dt_incl_act_lst, dt_incl_ext_num, dt_exc_ext_num.

Stability: num_sta, dt_start_ext_los, dt_end_ext_los, dt_stabil.

Inclusion: rea_incl, dt_incl, dt_end_act_duty.

Exclusion: rea_excl, dt_excl, cause_death.

Mil organization (Unit): abbrev, name, pred_rank_spe, eff_rank_spe, local, reg_com, majcom.

3.3.1 Identification of Entities

The first step in building the ERM for the personnel headquarters was to identify the entities in the model. By using the similarities approach, it was observed that some groups of data elements, like personnel, could be easily transformed into an entity. However, some other groups, like flight, should be subdivided into more than one entity.

The following entities were identified in the ERM for the personnel headquarters:

Personnel
Course
Exclusion
Death
Active_list (Specialty)
Mil_organization (Unit)
Leaving
Medal
IFR_card
Promotion
Inclusion
Aviator

Appendix B presents a more detailed description about the identified entities.

Entities that cannot be uniquely identified by their own attributes and must be identified by their relationships with other entities, are called "weak" entities. The following weak entities were identified in the ERM for the personnel headquarters:

Flight
Ext_num
Nomination
Stability
Moving
Attachment
Non_duty
Designation

Appendix B presents a more detailed description about the identified weak entities.

3.3.2 Identification of Relationships

After identifying the Entities, the next step was to identify the relationships among those entities that would satisfy the user needs. In this step the type of relationship, i.e., one-to-one, one-to-many or many-to-many, was also identified.

The following relationships were identified in the ERM for the personnel headquarters:

RELATIONSHIPS	TYPE
personnel_leave	many-to-many
personnel_promotion	many-to-many
personnel_course	many-to-many
personnel_medal	many-to-many
personnel_inclusion	many-to-many
personnel_exclusion	many-to-many
personnel_specialty	many-to-many
personnel_rank_flight	many-to-many-to-many
aviator_ifr_card	one-to-one
aviator_ext_num	one-to-many
personnel_stability	one-to-many
personnel_unit	many-to-one
personnel_unit_moving	many-to-many-to-many
personnel_unit_nomination	many-to-many-to-many

personnel_unit_designation	many-to-many-to-many
personnel_unit_attachment	many-to-many-to-many
personnel_specialty_non_duty	many-to-many-to-many
unit_promotion_specialty	many-to-many-to-many
aviator_promotion_flight	many-to-many-to-many

The entity personnel related to unit has four types of relationships, moving, nomination, designation, and attachment. It differs mainly because of the situation of the military person in the next unit. Moving is used during a standard move of the person from one unit to another; it is the most frequent case. Nomination is used when the person will be a commandant of the next unit. Designation is used when the person will be designated as instructor at the next unit. Attachment is used when the person is in one unit as a student during a short period of time, usually less than three months.

The entity personnel has a relationship with unit where neither of the previous four cases applies, the difference is that this relationship shows only the current unit assigned to the person, instead of previous units. In this case there is a single relationship between personnel and unit, to answer questions like "Where is Capt. Passos ?".

Other relationships were created in order to allow for questions related to personnel and other entities, such as leave, promotion, etc.

Appendix B presents a more detailed description about the identified relationships.

3.3.3 Identification of Attributes

The next step in the identification process was to select the attributes (properties) for each entity and for each relationship.

Along with the identification of the attributes for each entity, the key attributes were also identified. Key attributes can be composed of one or more attributes whose values uniquely identify an entity.

Some attributes are not related to entities, but are related to the relationship between entities. For example dt_start_course is an attribute of the relationship between the personnel and course entities.

To identify the key attributes of a relationship between two or more entities, the procedure is to get the key attributes of each entity involved, which combined, will be the key attributes for the relationship.

Some problems may arise when dealing with weak entities, which, by definition, are entities that depend on the existence of another entity. In this case, to form a unique identifier are used together: the key attribute from the strong entity, which the weak entity depends on, plus a minimal subset of attributes from the weak entity.

These attributes from the weak entity are called "discriminators". An example of a weak entity is Flight, that depends on personnel, where year_ref and qua_ref discriminate the entity from others, when linked to some person. These cannot uniquely identify the entity, since another person can have the same year_ref and qua_ref.

Key attributes for entities and relationships are referred to as entity_identifiers and relationship_ identifiers (9:23). Appendix B, contains a complete list of the attributes for each entity and relationship, and the respective identifiers.

3.4 Personnel Headquarters' ER Diagram

As a final product of the Entity Relationship Model, an ER Diagram was built to graphically represent the model.

In the ER Diagram a rectangle represents an entity, a double rectangle represents a weak entity, an ellipse represents an attribute, and a lozenge (diamond) represents a relationship.

The notations M, N and P represent more than one occurrence of the entity in the relationship, and 1 represents only one occurrence of the entity in the relationship.

As an example more than one element of the entity personnel can take some course and, more than one course can be related to the same person.

Another example is personnel_stability, where only one element of the entity personnel can be assigned to some stability, but, since stability has more than one occurrence, this relationship is one-to many. The last example is aviator_ifr where each element of the entity aviator has only one ifr_card, in this case the relationship is one-to-one.

Appendix C shows the ER Diagram for the personnel headquarters' personnel database. The diagram was split into 3 figures to better show the existing relationships. The diagram only represents the relationships among the defined entities and weak entities. A complete ER Diagram with all attributes for each entity, weak entity, and relationship, can be also found in the Appendix C.

IV. Normalization Process

After the ER Diagram has been created (Appendix C), the next step towards creating a conceptual model is the normalization process.

The normalization process can be described as a process of grouping data elements into tables representing entities and their relationships (3:130).

During the normalization process the concepts applied to relational data models are used. These concepts are applied to the analysis of the data and relationships provided by the end users.

An important point is that the conceptual model developed using the relational data model need not be implemented using a relational database management system. Rather, the model can be used as a basis to develop a logical model that can be implemented using a relational, hierarchical, or network database management system.

In this chapter the transition between an ER Diagram to a relational model is presented. Relations are created from existing entities, weak entities, and relationships defined in the ER Diagram. Once those relations have been created, they will be analyzed to ensure that they meet the requirements of fourth normal form.

4.1 Creating Relation from ER Diagram

A relation is a two dimensional table that has rows and columns, a row in the relation is called a tuple, a column is called an attribute of the relation.

Each relation can not have two equal tuples, so, in each pair of tuples, there must exist at least one attribute whose value is different.

Primary key is the set of one or more attributes that allow us to uniquely identify some tuple in a relation.

Candidate key is also the set of one or more attributes that allow us uniquely identify some tuple in a relation, the difference between both is that a primary key is the candidate key selected by the database designer to be the principal means of identifying tuples within a relation.

4.1.1 Entity --> Relation

Given an entity, the process to transform the entity into a relation, is to get the key attribute, or set of attributes, and to transform it into the primary key for the relation, and the nonkey attributes of the entity into attributes of the relation.

As an example the entity course, with attributes type_course, area_course, level_course, and name_course, was transformed into relation course, as can be seen in Figure 3.

In the relation course, type_course and name_course are candidate keys, but, since type_course is smaller than name_course, it was selected to be the primary key. The length of the key is an important factor to be considered in a primary key selection for two reasons: smaller information causes fewer typing errors than large information, and, since the primary key is the index for the relation, the access to a small index is faster than a larger index.

course

*type_course	area_course	level_course	name_course
EX-001	45	PG	COMPUTER SYSTEM GRADUATE
ECEMAR	30	AM	AIR FORCE COMMAND
EAOAR	30	AM	OFFICER IMPROVEMENT

Figure 3 - Relation Course (subset)

4.1.2 Weak Entity --> Relation

The process to transform a weak entity into a relation is to get the discriminator or set of discriminators of the weak entity and the key attributes of the entity on which the weak entity is dependent, to create a primary key for the relation. The nonkey attributes of the weak entity remain the nonkey attributes in the relation.

As an example, the weak entity ext_num, which is the situation where the aviator is placed if he has any health problem that does not allow him to fly, is dependent on entity aviator. In this case, the key attribute of aviator, which is rec_num, is combined with num_ext_num, which is the discriminator of ext_num, to form the primary key for the relation ext_num. The nonkey attributes dt_incl_ext_num and dt_excl_ext_num, will be the nonkey attributes for the relation, as can be seen in Figure 4.

ext_num

*rec_num	*num_ext_num	dt_incl_ext_num	dt_excl_ext_num
000018317100	1	120771	
000017153201	1	130870	140982
821123452341	1	130582	151283
821123452341	2	191084	
801020304050	1	100280	
811001200230	1	100183	280385

Figure 4 - Relation Ext_num

4.1.3 Relationship --> Relation

The process to transform such relationships into relations is getting the key attribute of each entity to form the primary key for the relation, and the nonkey attributes of the relationship became nonkey attributes of the relation.

As an example, the relationship personnel_promotion has rec_num as a key attribute of personnel, and rank as a key attribute of promotion. These two will become the primary key for the relation per_pro, and the nonkey attributes of the relationship, dt_pro and crit_pro, will became nonkey attributes of the relation, as can be seen in Figure 5.

per_pro

*rec_num	*rank	dt_pro	crit_pro
000018317100	MJ	241285	M
000018317100	CP	241278	A
000018317100	1T	300476	A
820112167902	2T	241282	A

Figure 5 - Relation Per_pro

4.2 Normal Forms

Normalization theory is built around the concepts of normal forms. A relation is said to be in a particular normal form if it satisfies a certain specific set of constraints (15:362).

```
+-----------------------------------------------------------+
| Universe of relations (normalized and unnormalized)       |
| +-------------------------------------------------------+ |
| | 1NF relations (normalized relations)                  | |
| | +---------------------------------------------------+ | |
| | | 2NF relations                                     | | |
| | | +-----------------------------------------------+ | | |
| | | | 3NF relations                                 | | | |
| | | | +-------------------------------------------+ | | | |
| | | | | BCNF relations                            | | | | |
| | | | | +---------------------------------------+ | | | | |
| | | | | | 4NF relations                         | | | | | |
| | | | | | +-----------------------------------+ | | | | | |
| | | | | | | PJ/NF (5NF) relations             | | | | | | |
| | | | | | +-----------------------------------+ | | | | | |
| | | | | |                                       | | | | | |
| | | | | +---------------------------------------+ | | | | |
| | | | |                                           | | | | |
| | | | +-------------------------------------------+ | | | |
| | | |                                               | | | |
| | | +-----------------------------------------------+ | | |
| | |                                                   | | |
| | +---------------------------------------------------+ | |
| |                                                       | |
| +-------------------------------------------------------+ |
|                                                           |
+-----------------------------------------------------------+
```

Figure 6 - Normal forms (15:363).

Figure 6 shows the existing normal forms that have been defined so far. For the purpose of this thesis effort, the normalization process goes until fourth normal form (4NF), the most commonly used normalization level.

Codd (11) originally defined First (1NF), Second (2NF), and Third (3NF) Normal Forms. Boyce/Codd Normal Form (BCNF) was defined also by Codd (12) as a new 3NF, because he discovered certain inadequacies in his original 3NF.

Subsequently, Fagin (16;17) defined fourth normal form, and Projection-Join Normal Form (PJ/NF), also considered a fifth normal form (5NF).

From Figure 6 it can be seen that all normalized relations are in 1NF, some in 1NF are also in 2NF, some in 2NF are also in 3NF, and so on until 5NF. The database designer should look for a design achieving the highest normal form possible.

4.3 Functional Dependencies

Two primary purpose of databases are to attenuate data redundancy and enhance data reliability. Any a priori knowledge of restrictions or constraints on the possible sets of data has considerable usefulness in reaching these goals.

Data dependencies are one way to formulate such advantageous knowledge, and the functional dependency is one type of data dependency (23:42).

Given a relation course, the attribute area_course is functionally dependent on attribute type_course, that can be expressed in symbols as:

```
type_course --> area_course
```

This can be read as type_course functionally determines area_course, and is valid if and only if each value in type_course has associated with it precisely one value in area_course.

Following the same pattern, the following notation can be expressed in symbols as:

```
type_course --> area_course
type_course --> level_course
type_course --> name_course
```

or, another way to express the same notation is:

```
type_course --> (area_course, level_course,
                 name_course)
```

Is important to note that if attribute type_course is a candidate key of the relation course, in particular, if it is the primary key, then all attributes of the relation course must necessarily be functionally dependent on type_course.

It is also important to note that based on the definition of functional dependency, there is no requirement that some attribute which functionally determines others, has to be candidate key of the relation.

As an example, the attributes dt_pro, and crit_pro are fully functionally dependent on attributes rec_num and rank of the relation per_pro, if they are functionally dependent on rec_num and rank, and not functionally dependent on any proper subset of rec_num and rank.

This notation can be expressed in symbol as:

rec_num, rank --> (dt_pro, crit_pro)

The relation per_pro would not be a fully functional dependence relation if were expressed as:

rec_num --> dt_pro, crit_pro

or

rank --> dt_pro, crit_pro

or any combination where the subset of rec_num and rank (i.e., rec_num alone or rank alone), determine any one of the attributes dt_pro, or crit_pro.

4.4 First, Second, and Third Normal Forms

First, second, and third normal forms are treated together because they were the original forms defined by Codd. BCNF and fourth normal form will be treated separately in later sections.

4.4.1 First Normal Form (1NF)

A relation R is in first normal form (1NF) if and only if all values in the domain A are atomic for every attribute A in R, that is, the values in the domain are not lists or sets of values or composite values (23:96). An atomic value is an attribute which contains a value that can not be divided.

As an example, suppose that in the relation personnel the interest on the attribute dt_birth was only in the month or year a person was born.

personnel

rec_num	dt_birth
000018317100	291048
000018534504	180247
821124326501	260952

In this case, the relation would not be in 1NF, since the deal is with part of a value. To be in 1NF the relation must be changed.

personnel

rec_num	day_birth	mo_birth	year_birth
000018317100	29	10	48
000018534504	18	02	47
821124226501	26	09	52

It is important to note that this is not true in the relation personnel, where the entire date is necessary, not only the month and day like in the example.

By using only first normal form, certain anomalies can arise in the database, and to avoid those anomalies, the normalization process has to go further.

The first anomaly that can arise is the insertion anomaly. Because there is no relation that can be used as a good example for this anomaly, an hypothetical relation, called "percourse", was created combining some attributes of the relation personnel and some of the relation course.

percourse

rec_num	tag_name	type_course	name_course
000018317100	mussato	EAOAR	OFFICER IMPROVEMENT
000018317100	mussato	AFA	AIR FORCE ACADEMY
000018317100	mussato	ECEMAR	AIR FORCE COMMAND
000018512304	oliveira	ECEMAR	AIR FORCE COMMAND
000018512304	oliveira	AFA	AIR FORCE ACADEMY
000018512304	oliveira	CPI	INSTR. PREPARATION
791236572904	da silva	TATICA	FLYING TACTICS
838741432503	braga	PILPRO	TEST PILOT

The insertion anomaly would occur if some course such as Computer System Graduate were inserted, and no person has the course. In this case the course has to be inserted leaving a key attribute, rec_num, blank. But, this is not possible, since it is violating the basic rule for relational model, trying to create a tuple without a primary key.

By using the same relation, a deletion anomaly would occur by deleting some person like "braga" from the relation and keeping his course. If braga is the only person who had the course, the person and his corresponding course must be deleted. If the intention is to keep the course, the same violation existing in the insertion anomaly, a tuple without a primary key, will occur.

The update anomaly would occur by trying to change the name of some course, with more than one person having the same course. In this case, the entire relation has to be searched to make the change for each person. This is a time consuming and possibly inaccurate operation.

The solution, in these cases, is to divide the relation into two, personnel and course. That allows insertion, deletion, and update on each relation separately without any complaint about anomalies.

personnel

rec_num	last name
000018317100	mussato
000018512304	oliveira
791236572904	da silva
838741432503	braga

course

type_course	name_course
EAOAR	OFFICER IMPROVEMENT
ECEMAR	AIR FORCE COMMAND
AFA	AIR FORCE ACADEMY
CPI	INSTR. PREPARATION
TATICA	FLYING TACTICS
PILPRO	TEST PILOT

4.4.2 Second Normal Form (2NF)

We say that a relation R is in the second normal form (2NF), if and only if it is in 1NF, and every nonkey attribute is fully functionally dependent on the primary key, that is, every nonkey attribute needs the full primary key for unique identification.

Relation percourse is an example of a relation that is in 1NF, but not in 2NF. Now, an additional attribute dt_start_course, will be added to the relation percourse. The primary key is composed of rec_num and type_course, such that the relation percourse can be expressed as:

```
rec_num --> tag_name, type_course, name_course
rec_num, type_course --> dt_start_course
type_course --> name_course
```

Figure 7 shows graphically the relation percourse.

```
                   +-------------+
 dt_               ¦             ¦
 start_ <----------¦ rec_num, ------------->  tag_name,
 course            ¦ type_course ¦            type_course,
                   ¦             \            name_course
                   +-------------+\
                                    ------>   name_course
```

Figure 7 - Functional dependencies
in the relation percourse

We see that name_course, type_course, and tag_name are not fully functionally dependent on the primary key, which is rec_num and type_course. Name_course is also functionally dependent on type_course, and type_course and tag_name, are also functionally dependent on rec_num.

In this case, the relation is not in the 2NF, the solution is to divide into two relations, percourse, and per_course, with the following dependencies:

percourse

```
rec_num --> tag_name, type_course, name_course
```

per_course

rec_num, type_course --> dt_start_course

Insertion, deletion, and update anomalies can also be found in the relation in 2NF, but in the example, the relation that contains such anomalies, percourse, was already analyzed in the previous section, and the comments remain the same.

4.4.3 Third Normal Form (3NF).

A relation is in third normal form (3NF), if and only if it is in 2NF, and every nonkey attribute is nontransitively dependent on the primary key (15:373).

Transitive dependencies can be explained as follows: If an attribute A determines an attribute B, and the attribute B determines attribute C, in this case a transitive rule is applied showing that attribute A determines also attribute C (33:218). In symbols it can be expressed as:

A --> B
B --> C

∴ A --> C

As an example, the relation percourse has the following dependencies:

rec_num --> tag_name, type_course, name_course

and

type_course --> name_course

In this case the relation has applied the transitive rule, since there exists the dependency

rec_num --> name_course

which is the result of a transitive dependency:

rec_num --> type_course
type_course --> name_course

∴ rec_num --> name_course

The relation that has been used so far, is not in 3NF, since it has the transitive dependency shown above. In order to make the relation in 3NF, the transitive dependency has to be eliminated.

One way to solve this problem, is to divide the relation percourse into two relations, personnel, and course, with the following dependencies:

personnel

rec_num --> tag_name, type_course

course

type_course --> name_course

Another way to solve this problem is to divide the relation percourse into three relations, personnel, course, and per_course, which is the relationship between those two relations, in this case it can be expressed as:

personnel

rec_num --> tag_name

course

type_course --> name_course

per_course

rec_num, type_course --> dt_start_course

As can be seen, the second solution is more applicable when there exist attributes that depend on the relationship between two relations. In this case, dt_start_course does not depends only on rec_num, neither only on type_course, but depends on both, i.e., depends on their relationship.

4.5 Boyce/Codd Normal Form (BCNF).

As mentioned early in section 4.4, BCNF was defined by Codd, because his original third normal form had some inadequacies. Those anomalies could be described as (15:374):

1 - multiple candidate keys,
2 - composite candidate keys and
3 - candidate keys overlapped.

Before the definition of the BCNF relation, it is convenient to introduce the term "determinant", that is, any attribute on which some other attribute is fully functionally dependent.

For example, in the relation percourse that was created in the previous section (Figure 7), attributes rec_num, type_course, and (rec_num, type_course), are all determinants.

A relation R is in Boyce/Codd normal form (BCNF), if and only if every determinant is a candidate key. It is interesting to note that in this definition, Codd starts using the term candidate key which was never used before in his first three definition of normal forms.

An example of a relation in BCNF, is the relation course, defined in the previous chapter.

```
course

     type_course --> name_course, area_course, level_course
     name_course --> type_course, area_course, level_course
```

Attributes type_course and name_course are both candidate keys (i.e. every course has a unique type_course, and also a unique name_course). In this case the relation course is in BCNF since the determinants, type_course and name_course, are candidate keys for the relation.

All relations described in Appendix D are in BCNF, since they are in 3NF, and all existing determinants are candidate keys.

4.6 Fourth Normal Form (4NF).

In order to define fourth normal form (4NF), it is convenient to introduce another term first. The term functional dependency was used early in previous sections. Now, another type of dependency is introduced, "multivalued dependency".

Given a relation R, attribute A multidetermines B, if, for each value of A there exists more than one corresponding value in B, or in other words, attribute B is multidependent on attribute A. In symbols it can be expressed as:

A -->-> B   (double-headed arrows)

In order to have a complete explanation of multivalued dependency, Date (15:384) said the following:

"Given a relation R with attributes A, B, and C, the multivalued dependency (MVD)

R.A -->-> R.B

holds in R if and only if the set of B-values matching a given (A-value, C-values) pair in R depends only on the A-value and is independent of the C-value. As usual, A, B, and C may be composite."

As an example of multivalued dependencies, the same example used by Maier is appopriate (23:123-124). The example uses the relation service with attributes Flight, Day_of_Week, and Plane_Type.

service

Flight	Day_of_Week	Plane_Type
106	Monday	747
106	Thursday	747
106	Monday	1011
106	Thursday	1011
204	Wednesday	707
204	Wednesday	727

As can be seen, there is no functional dependency Flight --> Day_of_Week or Flight --> Plane_Type in the relation service. However, the relation service can be decomposed into servday and servtype, as follows:

servday

Flight	Day_of_Week
106	Monday
106	Thursday
204	Wednesday

servtype

Flight	Plane_Type
106	747
106	1011
204	707
204	727

In this case there exists multivalued dependencies:

Flight -->-> Day_of_Service and
Flight -->-> Plane_Type,

since there is more than one value in the determined attribute for each value of the determinant attribute.

It is now appropriate to define fourth normal form which is related to multivalued dependency.

A relation R is in fourth normal form (4NF) if and only if there exists a nontrivial multivalued dependency in R, such as A -->-> B, then all attributes of R are also functionally dependent on A (15:385).

As an example, the relation service, with no functional dependencies is not in 4NF, but the decomposed relations servday and servtype, with multivalued dependencies, are both in 4NF.

All relations described in Appendix D are in 4NF, since they are in BCNF, and during their analysis no multivalued dependency was found.

Appendix D contains the analysis of all existing relations in the personnel database, until the Boyce/Codd Normal Form (BCNF).

V. Prototype Implementation

After the database has been designed, the next step in the software life cycle is the implementation phase (18:38). This chapter shows the implementation of a prototype of a personnel database for the Brazilian Air Force.

5.1 Prototype Considerations

The decision to develop a prototype was done considering the following reasons:

1 - the prototype can illustrate input data formats, messages, and interactive dialogues for the user.

2 - it is a valuable mechanism for explaining various processing options to the user, and for gaining better understanding of the user needs.

3 - it is a "complete system" that can easily be converted to the environment used in Brazil.

4 - the system was analyzed and designed without user interaction. The prototype will make possible a review of analysis and design phases before its full implementation.

In order to avoid misunderstanding about the prototype, it should be noted that a prototype is a "complete system", to be used as a tool to help the designer, during the development of the system. The prototype will not be used when the real system is implemented.

A prototype, typically, exhibits limited functional capabilities, low reliability, and/or inefficient performance (18:49-50). This prototype reflects these limitations. However, the main goal was achieved, to implement a system with the major functions - inclusion, modification, exclusion, and selection.

5.2 Selected Relations

The criteria for selecting the relations that would be included in the prototype was to have a minimum set of relations that could represent the entire system.

Using this idea, the decision was based on selecting the relations considered most accessed by the personnel users: moving, nomination, designation, and attachment. These relations are used mostly during the assignment of a person to another unit.

The complete list of all relation and their attributes, are listed below:

RELATION	ATTRIBUTES
personnel	recnum, tname, stbirth, dtexph, unith, crank, cactlist, cunit.
unit	abbrev, uname, local, regcom, majcom.
aviator	recnum, tname.
flight	recnum, rank, yearef, quaref, p1diu, p2diu, p1noc, p2noc.
promotion	rank.
specialty	actlist.

inclusion	reaincl, sumincl.
moving	recnum, abbrev, dtmov, dtpres, dtdetach, situ.
nomination	recnum, abbrev, dtnom, dtexo, sitn.
desig	recnum, abbrev, dtdes, dtwaiver, sitd.
attach	recnum, abbrev, dtsatt, dteatt, sita.
perpro	recnum, rank, dtpro, critpro.
perspe	recnum, actlist, dtactlist.
perinc	recnum, reaincl, dtincl.
unitprospe	abbrev, rank, actlist, pred, effect.

Figures 8 and 9 show the Entity Relationship Diagram for the prototype implementation. Only the entities involved and the relationships among them are shown.

Some relationships that appear on the ER Diagram (Figures 8 and 9) were not implemented as relations because they involve weak entities that already have the same attributes that exist in the relationships.

As an example, the relationship personnel-unit-moving (per_unit_mov), has recnum, abbrev, and dtmov as key attributes. Those key attributes also exist in the relation moving, since moving is a weak_entity that depends on personnel and unit. In this case, the relationship per_unit_mov no longer needs to be implemented.

The other case of a relationship that was not implemented as relation is personnel-unit (per_unit). In this case, instead of implementing the relationship per_unit as a

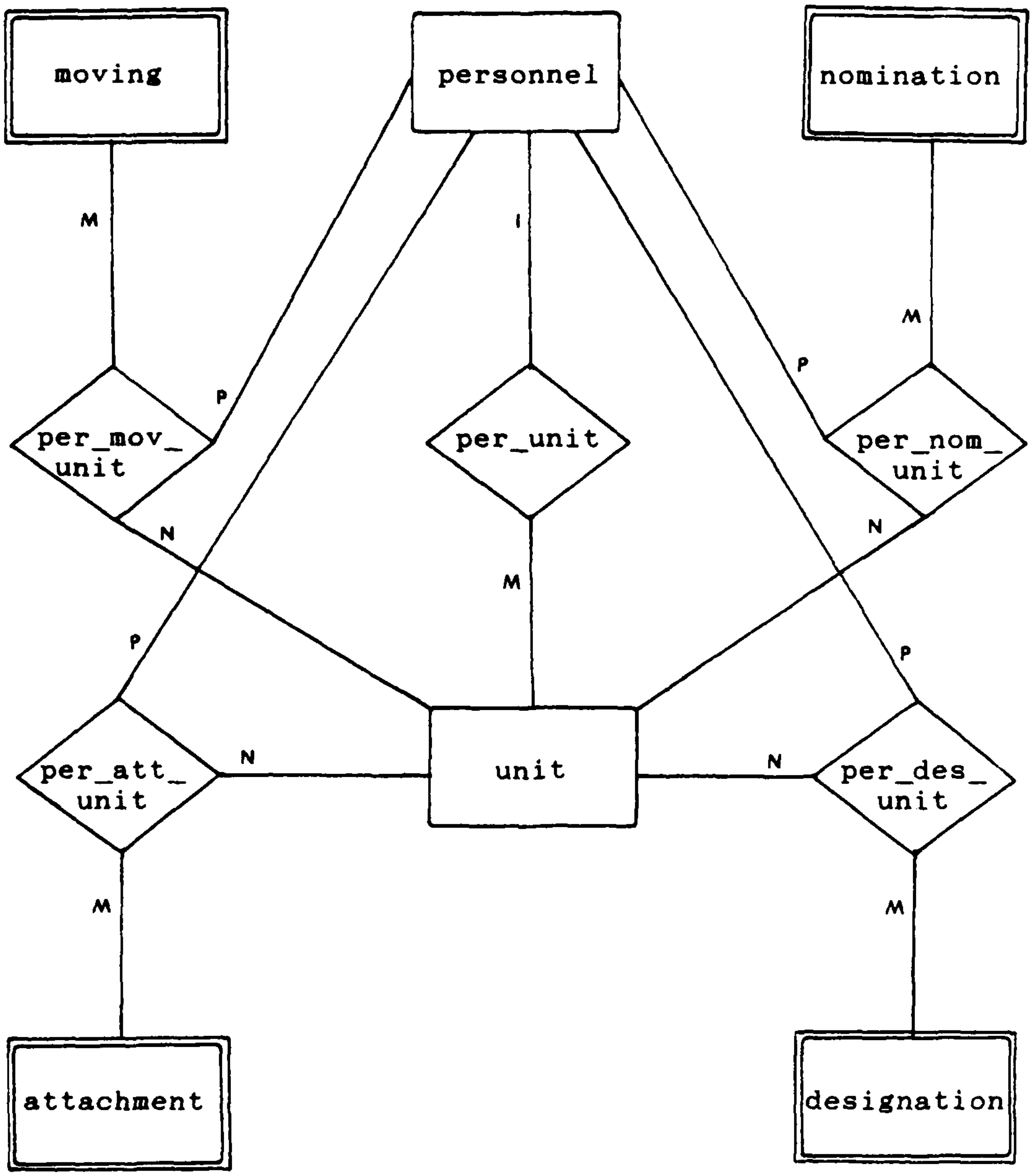


Figure 8 - ER Diagram for the prototype.

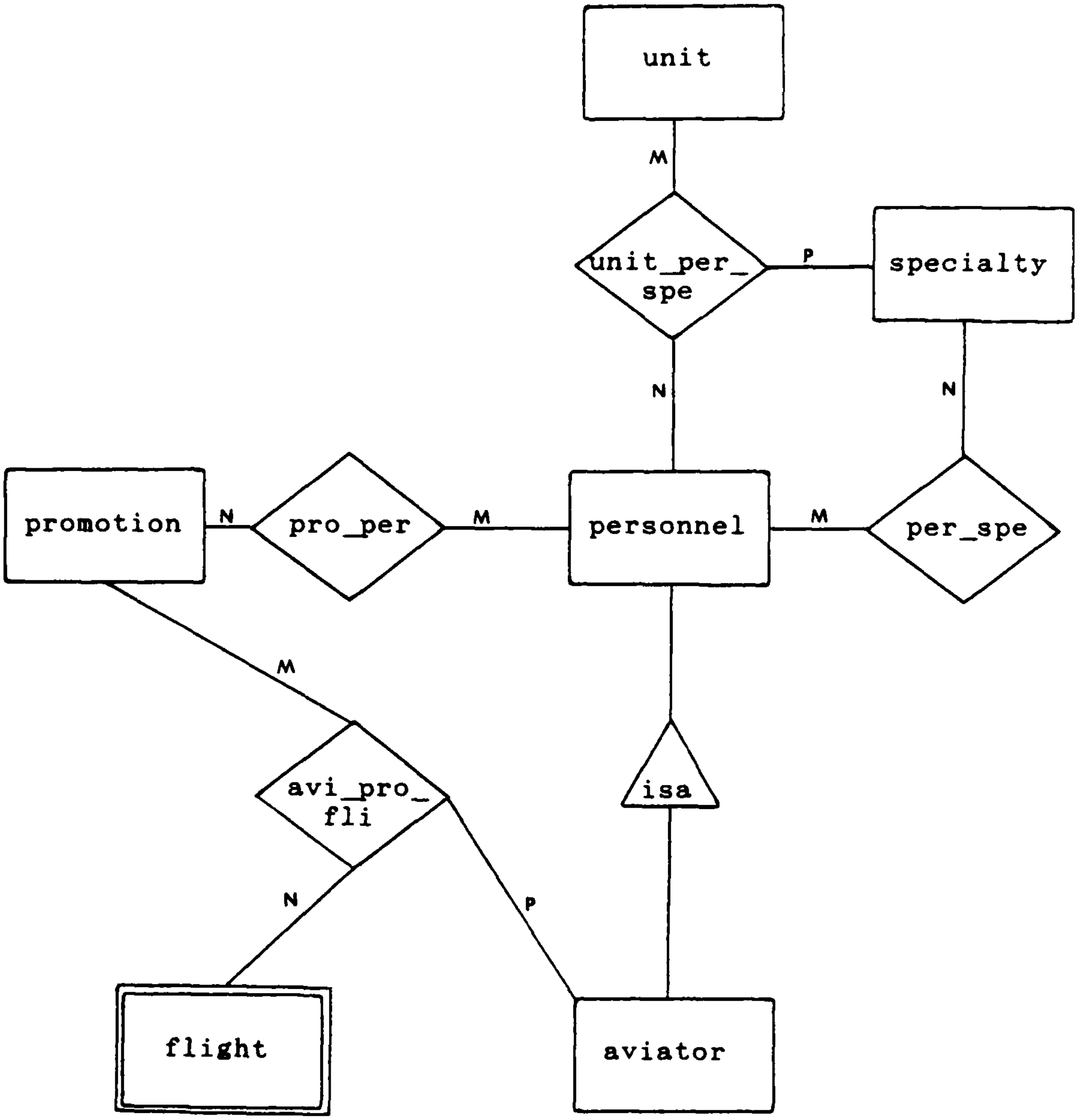


Figure 9 - ER Diagram for the prototype (cont).

relation, a primary key of the relation unit (abbrev) was added to relation personnel, as current unit (cunit), to allow necessary joins.

The following relationships, that exist on the ER Diagram, were not implemented as relations:

Per_unit_mov, Per_unit_nom,
Per_unit_des, Per_unit_att,
Per_unit and Avi_pro_fli.

The relation personnel includes three duplicated attributes, current rank (crank), current actlist (cactlist), and current unit (cunit). Cunit was added to relation personnel to allow natural join between personnel and unit. For reasons of performance, since most of the queries using personnel information, require rank and actlist, they were duplicated in the relation personnel to avoid constant join operations.

5.3 Selected Computational Environment

Among the existing DBMS at AFIT, one way to implement the prototype would be using INGRES, running on VAX, using the "C" language as a host language. This would be the natural way to implement the prototype, since this is the environment that was used during database courses.

The other option would be to use the DBMS ORACLE, that uses SQL as a query language, running on HARRIS 800, using Virtual Operating System (VOS), with COBOL as a host language.

The decision to use ORACLE was made because SQL can be more easily converted to the available system in Brazil. The host language was another advantage of ORACLE. Since COBOL is the host language to be used in Brazil, the use of the HARRIS and COBOL will also facilitate the transition.

5.3.1 Structured Query Language (SQL)

SQL, originally spelled SEQUEL, was first defined by Chamberlin and others at the IBM Research Laboratory in San Jose, California (15:55). A prototype implementation of the DBMS using SQL was built at the same laboratory, under the name "System R" (2). The results were very encouraging and System R is now known as DB2, SQL/DS. This is the computational environment that will be available in the Brazilian Air Force.

The basic structure of the SQL expression consists of three types of clauses: select, from, and where (22:71-80).

Select corresponds to the project operation of the relational algebra. It is used to get the desired attributes into the result of a query.

From is a list of relations to be used during the execution of the SQL expression.

Where corresponds to the select operation of the relational algebra. It consists of a predicate involving attributes of the relations that appear in the from clause.

An example of a simple query, using one of the relations of the prototype, is: "Find tag name of a person with record number 000018317100 in the relation personnel".

```
SELECT TNAME
FROM PERSONNEL
WHERE RECNUM = "000018317100"
```

5.3.2 Common Business Oriented Language (COBOL)

The COBOL language was developed in 1959, when it was recognized that a common standard language would be preferable to the proliferation of the languages being used at that time.

The first ANS (American National Standard) version of COBOL appeared in 1968, and a revised version appeared in 1974 (1:33).

At the time COBOL was written, the use of terminals was very restricted, the basic input data was done by punched cards. Probably due to this fact, COBOL does not have much flexibility in dealing with video terminals, for example, not allowing to control the cursor position on the screen.

This type of constraint had impact on the screen definition and limited its flexibility. Instead of sending several sets of information to the user at the same time, the program became restricted to sending and receiving only one set of information at a time.

5.4 Decision Support System (DSS)

A Decision Support System (DSS) can be characterized as an interactive computer based system that helps decision makers to utilize data and models to solve unstructured problems. These concepts were first articulated in the early 1970's by Michael S. Scott Morton (30:4).

Sprangue and Carlson (30) presented some systems that they consider examples of the DSS approach. On the list of four of these systems, three contain information about "historical data". This seems to be a key feature of DSS systems. Thus it was decided, early in the design phase, to keep historical data for most of the information in the personnel database.

Sprangue and Carlson noted that there exists a controversy and difficulty with terms like DSS, MIS (Management Information Systems), and EDP (Electronic Data Processing). Those problems can be traced to the difference between an academic or theoretical definition, and a "connotational" definition (30:6).

Figure 10 shows the connotational view of the three approachs in a single organizational chart, defining their area of performance.

The basic characteristics of each term is as follows:

EDP is the function that focuses on the lower operational level of the organization, such as paperwork automation. This function concentrates mostly on data, storage, processing, and flows at the operational level. EDP also deals with integrated files for related jobs and summary reports for management.

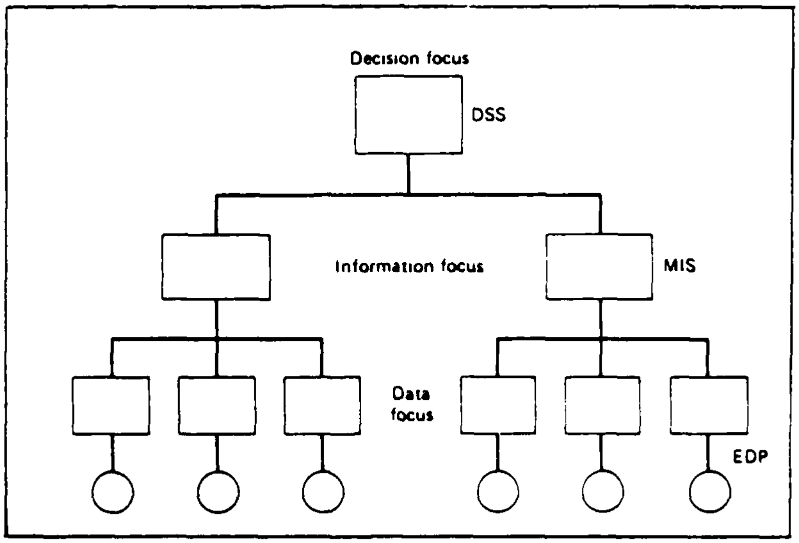


Figure 10 - The Connotational View (30:7).

The personnel database addresses this area with programs that allow inclusion, modification and deletion of all information within the system. This approach does not necessarily require a database system to support it. Instead, a conventional system with "master files" and magnetic tape could support this approach.

The MIS approach focuses on a higher level than EDP, with emphasis on integration and planning of the information systems function. The MIS approach also concentrates on information, rather than data, aimed at middle managers, dealing with structured information flows, inquiry and report generation. The MIS approach uses facilities of database. At this level the use of database tools, such as query languages, becomes more effective.

The personnel database addresses this area with some predefined queries that are frequently used by this level of management.

The DSS approach focuses on the highest level of information within the organization. It concentrates on top managers and executive decision makers, with emphasis on flexibility, adaptability, and quick response. At this point the database tools are no longer an option, but instead, a necessity.

The prototype uses SQL to show the top managers of the Brazilian Air Force that unstructured problems can be solved, or at least improved, by using this kind of tool, along with the available database with historical information.

The purpose of this thesis was not to build a DSS, because a strong interaction with the user has to be made. During the design such interaction could not be done. The main objective was to understand the DSS approach, and with the available tools existing in the relational DBMS, to implement the complete system for the personnel headquarters.

5.5 Prototype Storyboarding

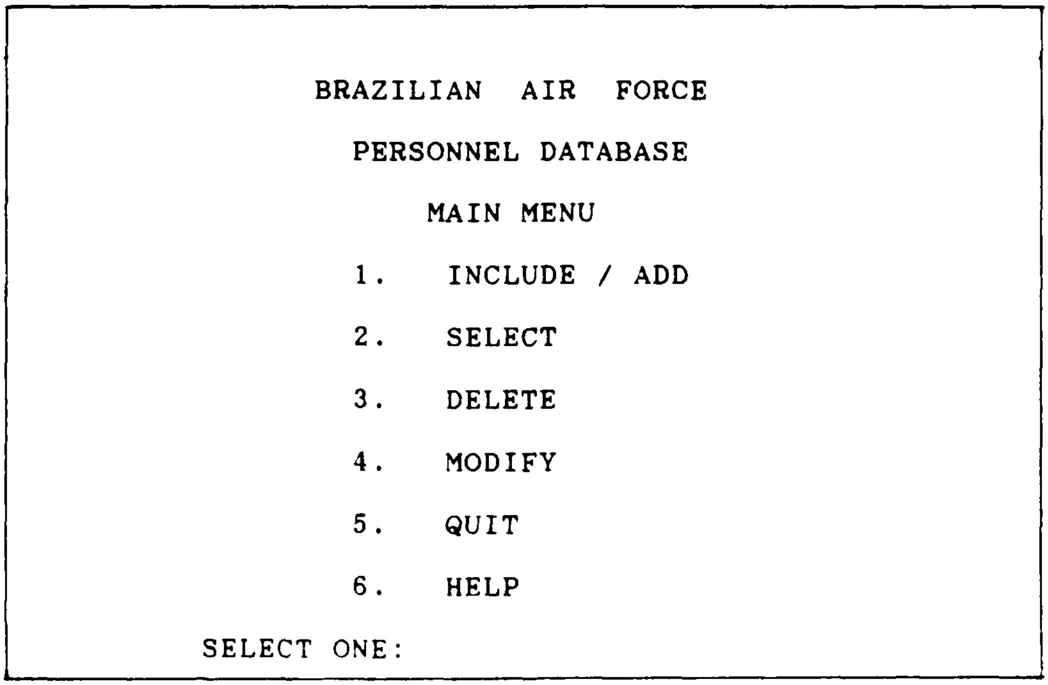
Early in this chapter some factors involved in the development of the prototype were presented, including ORACLE, SQL, COBOL, and DSS. This section combines the result of those factors, by showing the prototype storyboard, i.e., the set of screens designed, and gives some explanation about their use.

The program that controls the entire system is named "MAINMENU". Appendix E contains a printout of the entire program.

MAINMENU controls the system by calling subprograms to execute each specific function. Appendix E contains also a printout of one of this subprograms, named "SUBSEL", along with a list of the variable used in all subprograms. These variables are kept in a library and included in the program thru a COBOL command called COPY.

5.5.1 Main Menu

Figure 11 shows the first screen that appears when the system is started.



```
                BRAZILIAN  AIR  FORCE
                  PERSONNEL DATABASE
                      MAIN MENU
                  1.   INCLUDE / ADD
                  2.   SELECT
                  3.   DELETE
                  4.   MODIFY
                  5.   QUIT
                  6.   HELP
     SELECT ONE:
```

Figure 11 - Main Menu

This screen allows the user to control the system, by selecting one of the available choices. At this time, the system sends a message that ORACLE is being logged on.

By choosing option 1, a program named "SUBINCL" is called to perform inclusion or addition of data into the database. By choosing option 2, a program named "SUBSEL" is called to perform selection of several predefined functions. By choosing option 3, a program named "SUBDEL" is called to perform deletion of some information that was incorrectly included into the database. By choosing option 4, a program named "SUBMOD" is called to perform some update on the database. By choosing option 5, the user terminates normally the program. By choosing option 6, a help menu is presented.

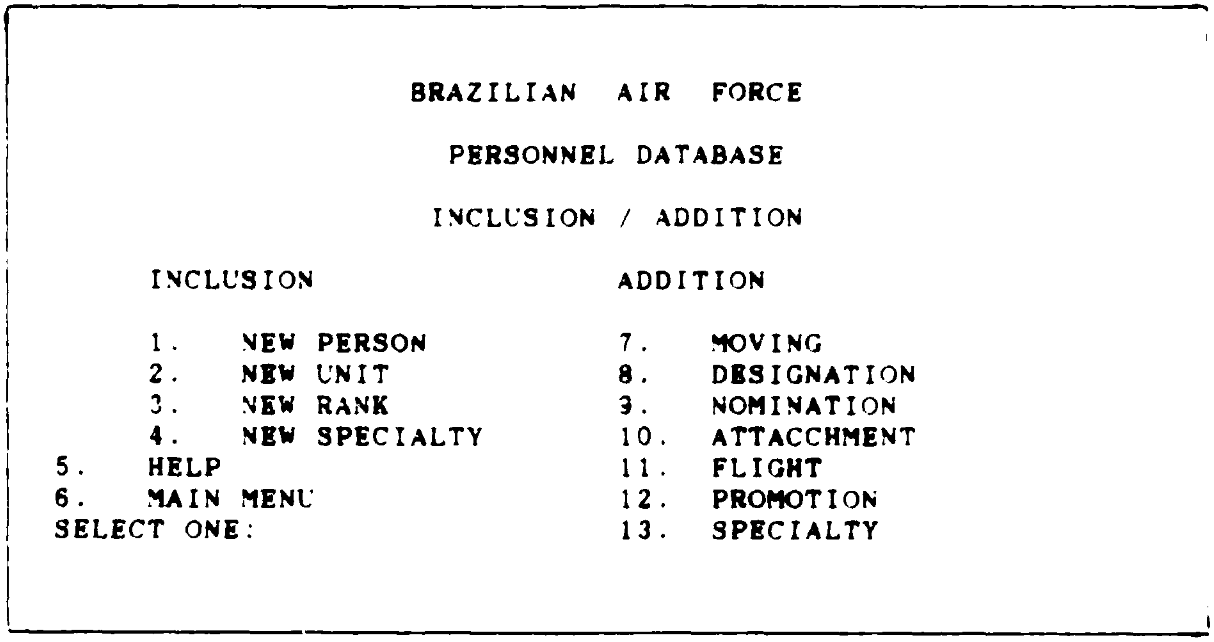
After the user selects one of the options, the system performs the desired function and returns to the main menu. If the user types an option not available, the system ask for another choice, and then if the user types again an option not available the system terminates.

5.5.2 Inclusion / Addition

Figure 12 shows the screen that appears when the user selects option 1 in the main menu.

This screen shows the options available in the Inclusion / Addition Menu. The screen is subdivided into three basic types of information, inclusion, addition and user aids.

On the left side, options 1 thru 4 are dedicated to inclusion of a new person, unit, rank or specialty. Options 5 and 6 are dedicated to user aids, HELP shows a short description about each available option, and MAIN MENU returns to the Main Menu screen.



```
                        BRAZILIAN  AIR  FORCE

                         PERSONNEL DATABASE

                         INCLUSION / ADDITION

      INCLUSION                          ADDITION

      1.   NEW PERSON                    7.   MOVING
      2.   NEW UNIT                      8.   DESIGNATION
      3.   NEW RANK                      9.   NOMINATION
      4.   NEW SPECIALTY                 10.  ATTACCHMENT
 5.   HELP                               11.  FLIGHT
 6.   MAIN MENU                          12.  PROMOTION
 SELECT ONE:                             13.  SPECIALTY
```

Figure 12 - Inclusion / Addition Menu

On the right side, options 7 thru 13 are dedicated to addition of moving, designation, nomination, attachment, flight, promotion, specialty.

The basic difference between inclusion and addition is as follows:

Inclusion has only one attribute as a primary key, and a prerequisite for an inclusion of a new person is that the key does not already exist in the database.

Addition has more than one attribute as a primary key, and an addition of a moving, with primary key composed of recnum, abbrev, and dtmov, means that recnum and abbrev already exist in the database, and dtmov is new in the database.

To present how an inclusion and an addition are being performed, let us select an inclusion of a new person and an addition of a moving as a legitimate sample for the process of inclusion and addition of information into the database.

By selecting option 1 (NEW PERSON) in the Menu, the system will start the dialog for the inclusion, asking:

Enter personnel RECNUM (12):

At this point the user has to type the record number of the new person, 12 characters long.

>801020304050 is an example of recnum.

RECNUM CHECKED is a message issued by the system telling the user that the recnum was checked and does not exist in the database.

Other messages can be issued when entering the recnum:

RECNUM MUST BE NUMERIC - TRY AGAIN, this means that the record number must be fully (12 characters) numeric.

RECNUM ALREADY EXIST - TRY AGAIN, this means that the record number that the user is trying to include already exist in the database.

TRY AGAIN means that the system starts the dialog asking again:

Enter personnel RECNUM (12): At this point, if the user for any reason intends to cancel the inclusion, just hit

<CR>, carriage return, and the system will ask:

```
RECNUM CAN NOT BE NULL
DO YOU WANT TO ABORT THIS INCLUSION {Y/N}?
```

At this point the user has to decide whether to abort the inclusion by typing "Y" or to continue the inclusion by typing "N".

This type of dialog where the system uses one line to ask a question and the user uses another line to answer the question, is done in the prototype because COBOL does not have control of the cursor on the screen, and during the implementation in the Brazilian Air Force this problem will be solved by using a communication program called Customer Information Control System (CICS) that interacts with COBOL when using video terminals.

Following is the remaining information needed to include a new person. Following the same pattern, the system asks a question and the user answer in other line.

```
Enter personnel STBIRTH (2):
>BA
Enter personnel TNAME (20):
>DA SILVA
Enter personnel UNITH (6):
>CEMAL
ABBREV CHECKED
Enter personnel DTEXPH (DD/MM/YY):
>12/12/87
Enter RANK (2):
>AP
RANK CHECKED
Enter CRITPRO {A/M/E/S}
>A
Enter ACTLIST (8):
>AV
ACTLIST CHECKED
Enter REAINCL (6):
>INCORP
```

```
REAINCL CHECKED
Entei DTINCL (DD/MM/YY):
>12/01/80
Enter Unit ABBREV (6):
>UNIFA
ABBREV CHECKED
```

As can be seen, every date has to be entered in the format DD for day, MM for numerical month, and YY for the last two digits of the year. Every data item that needs to be checked has a message issued by the system telling the user that the information was CHECKED. Every data item with choices between brackets ({}), such as CRITPRO, has to be answered using one of the options within the brackets.

When all the information needed to include a person has been supplied, the system continues asking:

```
DATA LOOK OK TO BE INCLUDED
CONTINUE THE PERSONNEL INCLUSION {Y/N}?
>Y
DA SILVA            INCLUDED IN THE PERSONNEL DATABASE
DA SILVA            INCLUDED IN THE AVIATOR RELATION
```

The objective of the question is that after this point a new person was included in the database, and this is the last opportunity to cancel, for any reason, the inclusion. The last two messages informs the user that a person was included in the personnel database, and also in the relation aviator when actlist is equal to "AV".

By selecting option 7 (MOVING), the system starts the dialog for addition of a moving asking:

```
Enter personnel RECNUM (12):
>801020304050
RECNUM CHECKED
```

```
Enter Unit ABBREV (6):
>AFA
ABBREV CHECKED
ENTER DATE MOVING (DD/MM/YY):
>06/06/81
```

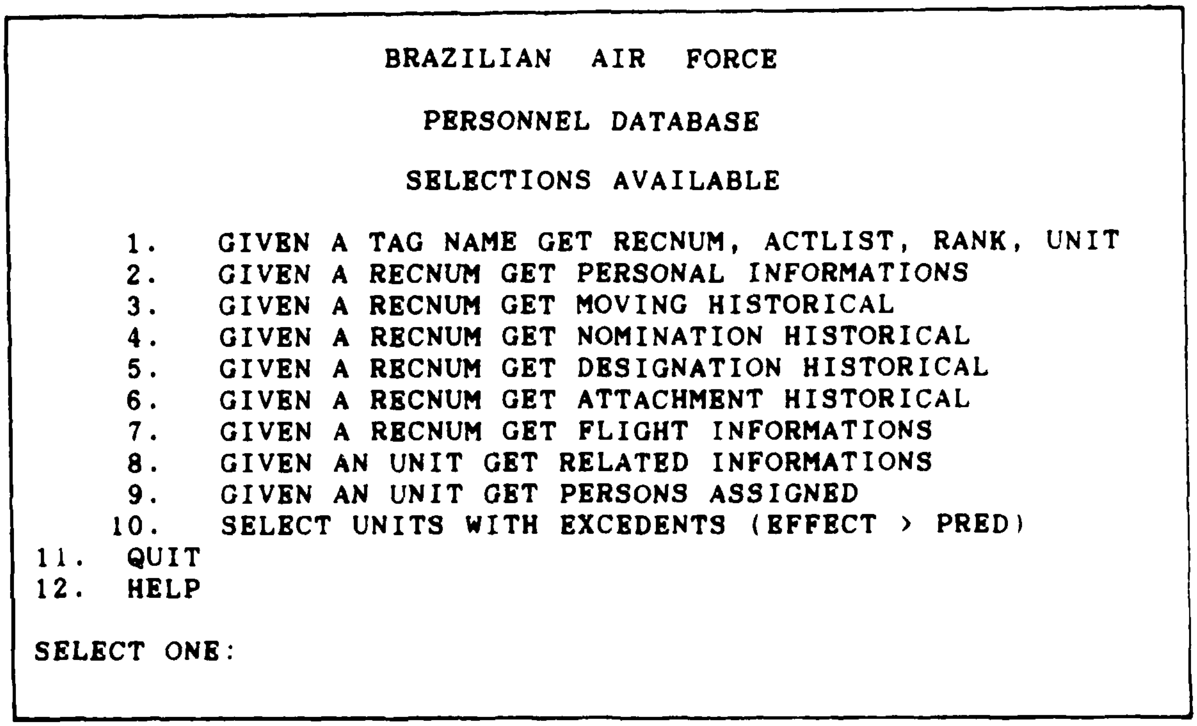
To move a person from one unit to another, the system needs the person, by asking for the RECNUM, check if the person exists on the database, issuing a message RECNUM CHECKED if he exists. Another item needed is the unit to be moved to, by asking for Unit ABBREV, checking for existence, and issuing a message ABBREV CHECKED. The last data item needed is the date of moving.

After all needed information is supplied, the system checks to see if such moving already exists. If not, the system will ask for continuing the addition and processes the moving, updating current unit in the relation personnel and adding one to the number of effective persons in the relation UNITPROSPE.

5.5.3 Select

Figure 13 shows the screen that appears when the user selects the option 2 in the main menu.

This screen shows the selections available for the user. Most of the information available in the prototype can be recovered by using these selections. Options 11 and 12 are the user aids that work like the Inclusion / Addition Menu.



```
                    BRAZILIAN  AIR  FORCE

                     PERSONNEL DATABASE

                    SELECTIONS AVAILABLE

      1.   GIVEN A TAG NAME GET RECNUM, ACTLIST, RANK, UNIT
      2.   GIVEN A RECNUM GET PERSONAL INFORMATIONS
      3.   GIVEN A RECNUM GET MOVING HISTORICAL
      4.   GIVEN A RECNUM GET NOMINATION HISTORICAL
      5.   GIVEN A RECNUM GET DESIGNATION HISTORICAL
      6.   GIVEN A RECNUM GET ATTACHMENT HISTORICAL
      7.   GIVEN A RECNUM GET FLIGHT INFORMATIONS
      8.   GIVEN AN UNIT GET RELATED INFORMATIONS
      9.   GIVEN AN UNIT GET PERSONS ASSIGNED
     10.   SELECT UNITS WITH EXCEDENTS (EFFECT > PRED)
 11.  QUIT
 12.  HELP

 SELECT ONE:
```

Figure 13 - Selections Menu

By selecting option 1 the following sequence is shown:

```
ENTER TAG NAME (20):
>DA SILVA

     RECNUM        ACTLIST       RANK       UNIT

801020304050         AV           AP        AFA
```

This type of selection is very useful to the user because most of the selections require RECNUM, and in the case that the user does not know the RECNUM, he can use this selection to get the information, along with other information that allows him to make sure that the correct person was selected.

By selecting option 2 the following sequence is shown:

```
        ENTER RECNUM (12):
        >801020304050

TAG NAME        STBIRTH UNITH   DTEXPH    RANK  ACTLIST  UNIT

DA SILVA          BA    CEMAL  12-DEC-87  AP      AV     AFA
```

With this selection the user has access to most of the personal information related to a given person. This type of selection is useful to identify the correct person before taking some action such as moving, promotion, etc..

Options 3 thru 6 display historical information about moving, nomination, designation, and attachment. Only option 3 (MOVING) will be presented, as a sample of this type of selection.

By selecting option 3 the following sequence is shown:

```
ENTER RECNUM (12):
>801020304050

ABBREV      DTMOV        DTPRES     DTDETACH      SITU

UNIFA     12-JAN-80    20-JAN-80   01-JUN-81      EFF
AFA       06-JUN-81    10-JUN-81                  EFF
```

With this selection the user has traced all the units assigned for some person. ABBREV is the unit, DTMOV is the date of moving, DTPRES is the date of presentation of the person in the unit, DTDETACH is the date that the person left the unit, SITU is the situation of the person in that unit, EFF means effective in the unit, i.e., the person is not an instructor, nor a commandant, etc..

Following the same pattern, option 7 shows flight information given a RECNUM, option 8 shows unit information given an ABBREV, and option 9 shows persons assigned given an ABBREV. Those selections will not be presented here because they follow the same pattern as the selections already presented.

The last selection available is also useful for the user to select the units with excedents of person, in other words, with a more effective person than predicted. This information is always required during the process of moving.

By selecting option 10 the following sequence is shown:

```
UNIT      RANK      ACTLIST    EFFECTIVE      PREDICT

CCA-RJ     MJ         AV         00006         00005
CCA-RJ     1T         AV         00006         00005
CCA-RJ     2T         AV         00006         00005
CCA-RJ     CP         AV         00006         00005
CCA-RJ     MJ         INT        00001         00000
UNIFA      CP         AV         00006         00005
AFA        AP         AV         00004         00000
UNIFA      AP         AV         00001         00000
```

For each unit is predicted, each year, the number of persons, within each rank and each actlist, assigned to the unit. This is part of Moving Plan, which is a plan of assignments for each unit, and the information of which unit has excedents is very important for this planning. It is important to notice that this selection is dynamic, because persons are moved daily.

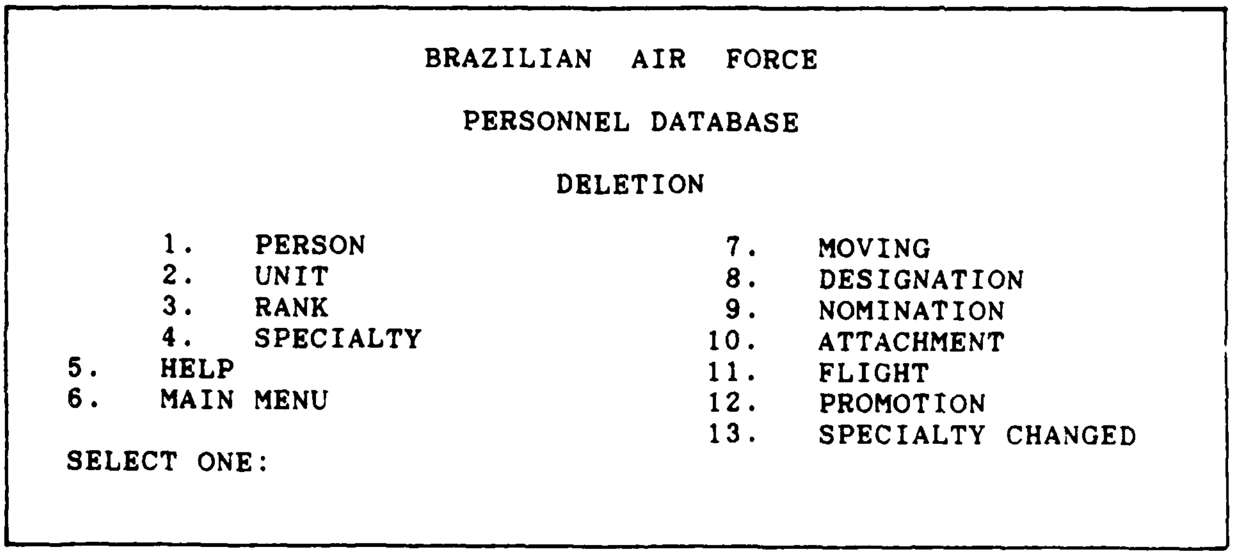
With these selections available the user will have a better understanding of the system, making possible a more accurate request of new selections for the implementation phase in the Brazilian Air Force.

5.5.4 Deletion

Figure 14 shows the screen that appears when the user selects option 3 in the main menu.

This screen shows the options available in the Deletion Menu. The screen is subdivided into two basic types of information: deletion and user aids.

On the left side, options 1 thru 4 are dedicated to deletion of a person, unit, rank or specialty. Options 5 and 6, are dedicated to user aids, HELP shows a short description about each available option, and MAIN MENU returns to the Main Menu screen.



```
                      BRAZILIAN  AIR  FORCE

                        PERSONNEL DATABASE

                             DELETION

     1.   PERSON                          7.    MOVING
     2.   UNIT                            8.    DESIGNATION
     3.   RANK                            9.    NOMINATION
     4.   SPECIALTY                      10.    ATTACHMENT
5.   HELP                                11.    FLIGHT
6.   MAIN MENU                           12.    PROMOTION
                                         13.    SPECIALTY CHANGED
SELECT ONE:
```

Figure 14 - Deletion Menu

On the right side, options 7 thru 13 are dedicated to deletion of moving, designation, nomination, attachment, flight, promotion, and specialty changed.

Deletion is used to remove some information that was included or added incorrectly in the database. If a person was included incorrectly, all information about that person must be removed from the database. But, if some unit was included incorrectly in the database, and a person was assigned to that unit, the procedure to remove the incorrect unit is to modify the incorrect assignment to a correct unit and to delete the incorrect unit.

By selecting option 1 (PERSON) in the menu, the following sequence will be shown:

```
ENTER RECNUM (12):
>801020304050
PERSONNEL CHECKED
CHECKING PERSON...
AVIATOR CHECKED
RECNUM IN AVIATOR
FLIGHT CHECKED
MOVING CHECKED
RECNUM IN MOVING
NOMINATION CHECKED
RECNUM IN NOMINATION
DESIGNATION CHECKED
ATTACHMENT CHECKED
PERPRO CHECKED
RECNUM IN PERPRO
PERSPE CHECKED
RECNUM IN PERSPE
PERINC CHECKED
RECNUM IN INCLUSION

PLEASE CHECK ABOVE WHICH RELATION HAS THIS RECNUM
THIS RECNUM WILL BE DELETED FROM THOSE RELATIONS

CONTINUE PERSON DELETION (Y/N)?
>N
```

During the process of deleting some person, all existing relations must be checked to ensure that the RECNUM is not there. If some relation contains the RECNUM a message is issued warning the user.

At the end of the check, a message is sent to the user warning about relations with the RECNUM. The user has to decide whether to continue with the deletion.

By selecting option 2 (UNIT) in the menu, the following sequence will be shown:

```
ENTER UNIT ABBREV (6):
>UNIFA
UNIT CHECKED
CHECKING UNIT...
CUNIT CHECKED
UNIT IN CUNIT (PERSONNEL)
UNIT CAN NOT BE DELETED
UNIT DELETION ABORTED
```

This is an example of what was said before about a unit to be deleted, and the unit has been assigned to a person. In this case, the system sends a message that the unit is in some relation and can not be deleted. The process to delete this unit is aborted. If the user needs to delete the unit, he first has to modify the relations that contain the unit, and start the process again. This type of situation can also occurs with RANK and SPECIALTY.

The remaining deletions follow the same pattern as MOVING. In this case the sequence for moving will be shown as a sample:

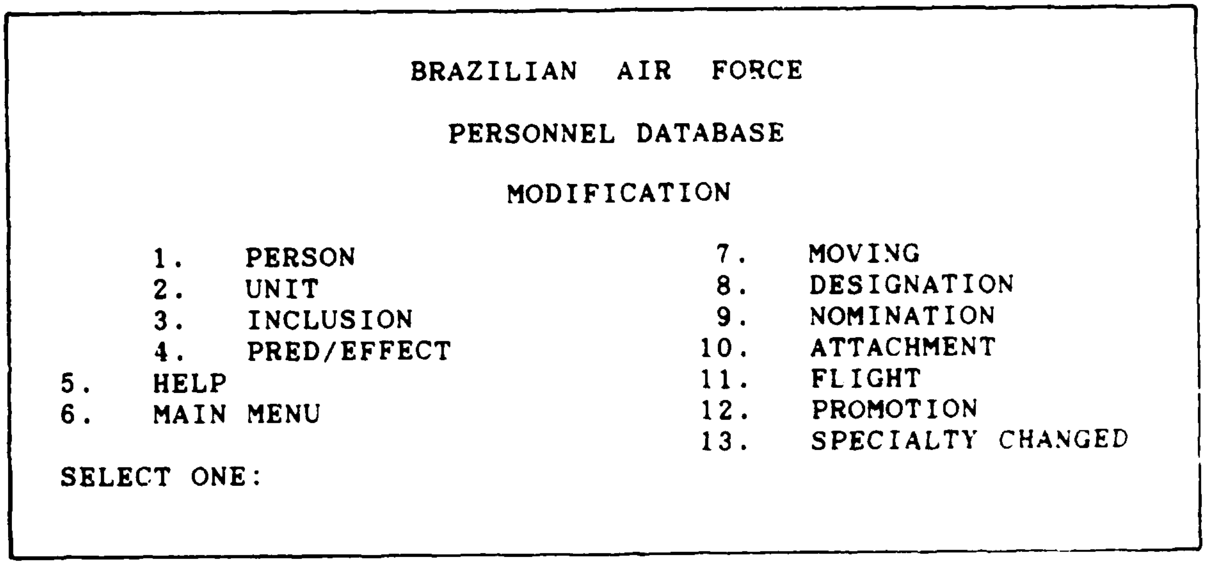
```
ENTER RECNUM (12):
>801020304050
PERSONNEL CHECKED
ENTER UNIT ABBREV (6):
>AFA
```

```
UNIT CHECKED
ENTER DATE OF MOVING (DD/MM/YY):
>06/06/81
DATA LOOK OK TO BE DELETED
CONTINUE MOVING DELETION (Y/N)?
>N
```

To delete a moving the required information is asked and checked before the deletion. Again the user is asked to continue the process or cancel, as a last chance before updating the database.

5.5.5 Modification

Figure 15 shows the screen that appears when the user selects option 4 in the main menu.



```
                      BRAZILIAN  AIR  FORCE

                       PERSONNEL DATABASE

                          MODIFICATION

     1.   PERSON                        7.   MOVING
     2.   UNIT                          8.   DESIGNATION
     3.   INCLUSION                     9.   NOMINATION
     4.   PRED/EFFECT                  10.   ATTACHMENT
5.   HELP                              11.   FLIGHT
6.   MAIN MENU                         12.   PROMOTION
                                       13.   SPECIALTY CHANGED
SELECT ONE:
```

Figure 15 - Modification Menu

This screen shows the options available in the Modification Menu. The screen is subdivided into two basic types of information: modification and user aids.

On the left side, options 1 thru 4 are dedicated to modification of a person, unit, inclusion or pred/effect. Options 5 and 6 are dedicated to user aids; HELP shows a short description about each available option, and MAIN MENU returns to the Main Menu screen.

On the right side, options 7 thru 13 are dedicated to modification of moving, designation, nomination, attachment, flight, promotion, and specialty changed.

Modification is mostly used to update the database, but can also be used to change information that was included or added incorrectly in the database.

The process of modifying some information in the database follows the same pattern for all relations. The user is asked to enter the key for the relation to be updated, and the system will ask, for each attribute, if he intends to modify that attribute, by typing "Y", or not, by typing either "N" or just hitting <CR>.

Because of the similarity, only modification of a person and pred/effect will be presented.

By selecting option 1 (PERSON) the following sequence will be shown:

```
ENTER RECNUM (12):
>801020304050
PERSONNEL CHECKED

RECNUM        TNAME    STBIRTH UNITH  DTEXPH    RANK  SPE UNIT
801020304050  DA SILVA   BA    CEMAL 12-DEC-87   AP   AV  AFA
```

```
MODIFY STATE BIRTH {Y/N}?
>
MODIFY UNITH HEALTH {Y/N}?
>
MODIFY DT EXP HEALTH {Y/N}?
>Y
ENTER DT EXP HEALTH (DD/MM/YY):
>15/12/88
DTEXPH UPDATED
MODIFY CURRENT UNIT {Y/N}?
>N
MODIFY CURRENT ACTLIST {Y/N}?
>
MODIFY CURRENT RANK {Y/N}?
>N
```

By selecting option 4 (PRED/EFFECT) the following sequence will be shown:

```
ENTER UNIT (6):
>CCA-RJ
ENTER RANK (2):
>MJ
ENTER ACTLIST (9):
>INT
PRED/EFFECT CHECKED
PREDICT IS 00000 - EFFECT IS 00001
MODIFY PREDICT {Y/N}?
>Y
ENTER PREDICT (5):
>00001
PREDICT UPDATED
MODIFY EFFECT {Y/N}?
>N
```

With Modification, the storyboard is finished, and the most important parts of the prototype were presented. As said before, a prototype is not a real system, but instead, a virtual system that will be easily implemented in the Brazilian Air Force.

5.6 DSS Queries

As a last part of the prototype development, some queries are presented, to show the potential use of the tool, that, together with the user, will make it possible to build a complete DSS.

A regular query involving dates is very difficult to be used inside some program, since dates can be changed for each query. In this case, a better solution is to use SQL queries, instead of creating a program to solve the problem.

This type of problem is typically unstructured, and the following sequence is a good sample of this type of query.

A typical question could be: "List all persons that were included in the B.A.F. within the period of 01-jan-76 and 01-jan-80".

A SQL query to recover this information from the prototype is as follows:

```
UFI>
>SELECT RECNUM, CRANK, TNAME

>FROM PERSONNEL

>WHERE RECNUM IN

>     (SELECT RECNUM

>     FROM PERINC

>     WHERE DTINCL BETWEEN '01-JAN-78' AND '01-JAN-80'
```

The result of the query would be

```
RECNUM       CR TNAME
------------ -- -------------------
801020304050 AP DA SILVA
000011112222 AP MARCOS
000999888777 AP TADEU
000000000001 AP ZACARIAS
```

Another example of how SQL could help the decision makers of the personnel headquarters, is answering some typical question such as: "List all persons, among all ranks and actlists, that in 01-jan-80 were captain".

The SQL query to recover this type of information from the prototype is as follows:

```
UFI>
>SELECT RECNUM, CRANK, CACTLIST, TNAME
>FROM PERSONNEL
>WHERE RECNUM IN

>    (SELECT RECNUM
>    FROM PERPRO
>    WHERE RANK = 'CP' AND DTPRO < '01-JAN-80' AND
>    RECNUM IN

>         (SELECT RECNUM
>         FROM PERPRO
>         WHERE RANK = 'MJ' AND DTPRO > '01-JAN-80'));
```

The answer to that SQL query could be:

```
RECNUM       CR CACTLIST  TNAME
------------ -- --------- --------------------
000016916900 TC ENG       PASCOAL

UFI>
```

These two queries presented some of the potential applications that will be used for the decision makers of the personnel headquarters. Initially, the purpose will be to generate several predefined queries, where the user must fill the spaces with changed values, such as date.

The important point is that the tool exists and can be used according to the needs of the decison makers, and this prototype was developed for that goal.

VI. Conclusion

This chapter presents the conclusion of the research, and makes some recommendations to improve the usage of the database for future research in this area.

6.1 Conclusion

A complete design of a large database is a difficult and time consuming task, even for persons with much experience in this area.

The personnel database for the Brazilian Air Force is now ready to be fully implemented. By using techniques such as E-R Modeling and Normalization Theory, the research done during this period gives a well defined baseline from which to proceed with full-scale implementation.

Research in the area of relational databases and the development of the prototype demonstrates the feasibility of this concept to be applied to future database designs in the Brazilian Air Force.

6.2 Recommendations

As a natural consequence of the developed research, the following recommendations are offered:

1 - The prototype, developed and implemented using COBOL under HARRIS 800, should be converted to the computation environment existing in the Brazilian Air Force;

2 - After being converted, the prototype should be

promptly implemented;

3 - With the prototype implemented, an evaluation of the entire system should be done by its users;

4 - After the evaluation, the complete system should be implemented, using a DSS approach;

5 - A preliminary study should be initiated to develop a distributed database system as a natural solution in this area, considering the following factors:

a) the users of the personnel database are mostly concentrated in two cities, Rio de Janeiro and Brasilia;

b) both cities have a computation center (CCA-BR and CCA-RJ) already installed, using the same operating system (DOS-VSE);

c) communications cost would be reduced, because the users will access local databases and,

d) distributed database technology has become more mature in the past years.

6 - COBOL as a host language should only be used for special applications because it requires a large number of lines of code to be used. As an example, the prototype has around 8,000 lines of code.

A recommended solution for some of these problems is to use another type of language, such as fourth generation languages during the implementation of the personnel database. This will reduce time, lines of code and make the system more user friendly.

The use of Fourth Generation Language (4GL) has increased in the past years because of the new tools and application development techniques that are being introduced.

Fourth Generation Languages vary greatly in power and capabilities. Some are merely query languages; oth:rs are report generators; and others can generate complete applications, and can be employed by the end users or system analysts.

Along with 4GLs, there exists a wide range of tools designed to increase productivity. Among several tools selected for The James Martin Report (24), are the following:

Very High-Level Procedural Languages: Tools that provide a well structured procedural language, that give results of <u>one</u> <u>tenth</u> of the time or less required by third generation language, such as COBOL or FORTRAN.

Distributed Microcomputer Support: Integrated micro/ mainframe support including, ideally, a version of the tool for both mainframe and distributed personal computer environments.

Database and Communication Support: Interfaces to widely used DBMSs and communication support facilities.

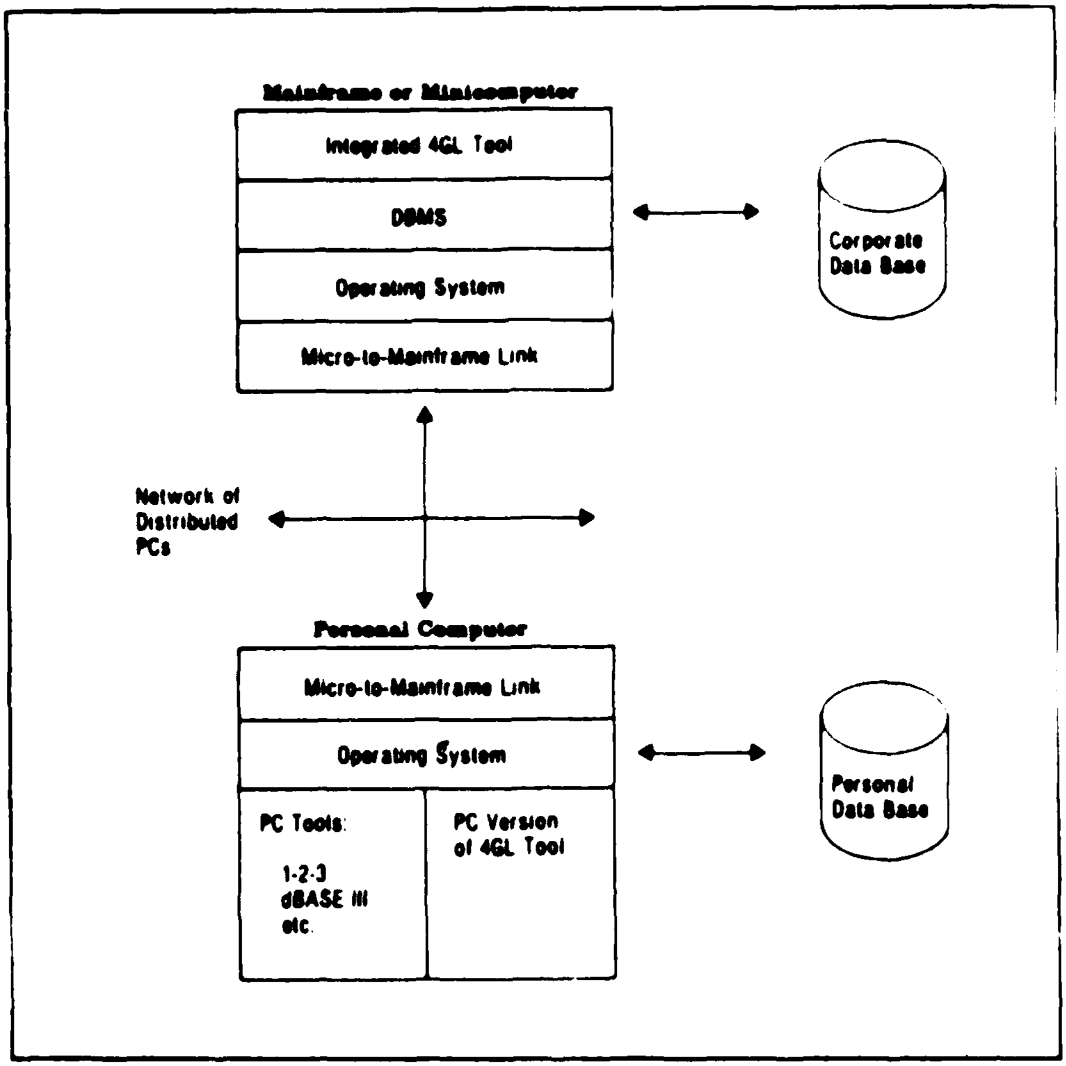


Figure 16 - Architecture for End-User Computing (24:4)

Figure 16 shows an Architecture for End-User Computing, by using a network of distributed personal computers. Under this environment the End-User could have his own applications in the personal computer and also could access the personnel database in the mainframe.

Figure 17 shows a more detailed picture of this environment, presenting also, as an example, some available software in the market.

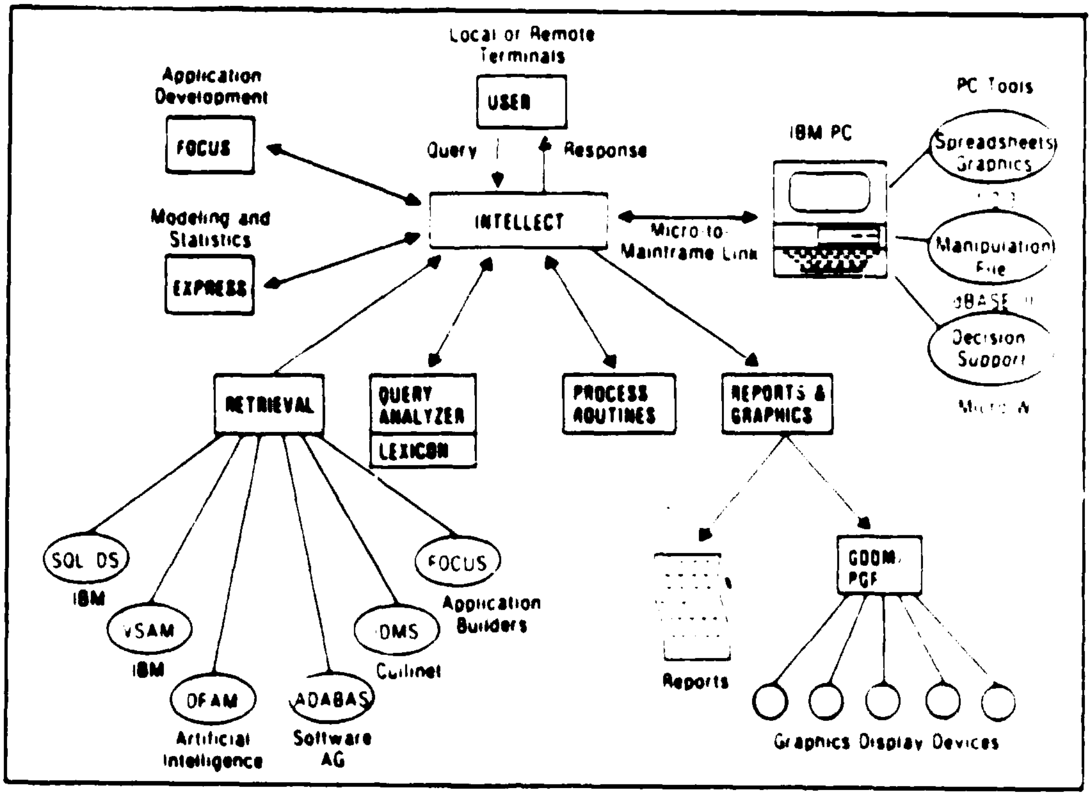


Figure 17 - Example of Integrated Facilities (24:12)

This is the environment recommended for the Brazilian Air Force. This environment represents an optimal solution for the development of applications using integrated facilities. Large databases stored in mainframes, such as personnel database, can be shared for several users and small databases stored in personal computers, can be locally used for its users.

Appendix A

Personnel Database Data Dictionary

After analyzing the Data Flow Diagram the data elements were identified, and this data dictionary was built in order to better explain the meaning of each data element.

CODE NAME PORTUGUESE NAME ENGLISH
Description
Domain (length - type)

IN01 NAT_INCL REA_INCL
Reason of inclusion in the BAF.
01 - numeric (code)

IN02 DAT_INCL DT_INCL
Date of inclusion in the BAF.
06 - numeric

IN03 RES_INCL SUM_INCL
Summary of the reason for inclusion in the BAF.
30 - alphanumeric

IN04 DAT_TERM_SVTA DT_END_ACT_DUTY
Date of end of active duty service.
06 - numeric

PE01 RC REC_NUM
Record number, uniquely identify the military in the BAF.
12 - numeric

PE02 NOME NAME
Full name.
64 - alphanumeric

PE03 NATURAL STATE_BIRTH
State of birth.
02 - alphanumeric (code)

PE04 DATA_NASC DT_BIRTH
Date of birth (dd/mm/yy).
06 - numeric

PE05 NOME_PAI FATHER_NAME
Father name from the birth certificate.
64 - alphanumeric

PE06 NOME_MAE MOTHER_NAME
Mother name from the birth certificate.
64 - alphanumeric

PE07 IDENT ID_NUM
Number of the identification card.
12 - numeric

PE08 SEXO SEX
Sex.
01 - alphabetic

PE09 CPF INC_TAX_NUM
Income tax number.
11 - numeric

PE10 PIS/PASEP SOC_SEC_NUM
Social security number.
11 - numeric

PE11 FUNSA MED_REC_NUM
Medical record number.
08 - numeric

PE12 NOME_GUERRA TAG_NAME
Tag name, used for short, instead of full name.
20 - alphabetic.

MO01 OM_DEST UNIT_DEST
Military organization of destination when moving.
03 - numeric (code)

MO02 DAT_MOV DT_MOV
Date of moving.
06 - numeric

MO03 DAT_APRES DT_PRES
Date of presentation in the new military organization.
06 - numeric

MO04 DAT_DESL DT_DETACH
Date of detachment of the previous military organization.
06 - numeric

MO05 OM_MOV UNIT_MOV
Name of the new military organization moving to.
03 - numeric (code)

MO06 SIT_OM SIT_UNIT
Situation of the military in the organization.
02 - alphanumeric (code)

MO07 RES_MOV SUM_MOV
Summary of the reason for moving.
30 - alphanumeric

DE01 OM_DESIG UNIT_DESIG
Military organization of designation.
03 - numeric (code)

DE02 DAT_DESIG DT_DESIG
Date of designation.
06 - numeric

DE03 DAT_DISPENSA DT_WAIVER
Date of waiver in the previous military organization.
06 - numeric

DE04 SIT_OM SIT_UNIT_DES
Situation of the military in the organization of designation.
02 - alphanumeric

DE05 RES_DESIG SUM_DESIG
Summary of the reason for designation.
30 - alphanumeric

NO01 OM_NOM UNIT_NOM
Military organization of nomination.
03 - numeric (code)

NO02 DAT_NOM DT_NOM
Date of nomination.
06 - numeric

NO03 DAT_EXO DT_EXO
Date of exoneration in the previous military organization.
06 - numeric

NO04 SIT_OM SIT_UNIT_NOM
Situation of the military in the organization of nomination.
02 - alphanumeric (code)

NO05 RES_NOM SUM_NOM
Summary of the reason for nomination.
30 - alphanumeric

AD01 OM_ADID UNIT_ATT
Military organization of attachment.
03 - numeric (code)

AD02 NAT_ADID REA_ATT
Reason of attachment.
02 - numeric (code)

AD03 DAT_INIC_ADID DT_START_ATT
Date of start the attachment.
06 - numeric

AD04 DAT_TERM_ADID DT_END_ATT
Date of end of attachment.
06 - numeric

AD05 SIT_OM SIT_UNIT_ATT
Situation of the military in the organization of attachment.
02 - numeric (code)

AD06 RES_ADID SUM_ATT
Summary of the reason for attachment.
30 - alphanumeric

LI01 TIP_LIC TYPE_LEAVE
Type of leave according to existing code.
01 - numeric (code)

LI02 NOME_LIC NAME_LEAVE
Name of the leave.
20 - alphanumeric

LI03 DAT_INIC_LIC DT_START_LEAVE
Date of start the leave (dd/mm/yy).
06 - numeric

LI04 DAT_TERM_LIC DT_END_LEAVE
Date of end of the leave.
06 - numeric

PR01 POSTO_GRAD RANK
Rank of the military.
02 - alphanumeric (code)

PR02 DAT_PROM DT_PROM
Date of promotion to the rank (dd/mm/yy).
06 - numeric

PR03 CRIT_PROM CRIT_PROM
Criterion of promotion according to existing code.
01 - alphabetic

FL01 ANO_REF YEAR_REF
Year of the flight (last two).
02 - numeric

FL02 TRIM_REF QUA_REF
Quarter of reference of the flight
01 - numeric

FL03 HS_DIU_1P_TRIM HS_DIU_1P_QUA

Hours flown during the day as first pilot in the quarter.
05 - numeric

FL04 HS_DIU_2P_TRIM HS_DIU_2P_QUA
Hours flown during the day as second pilot in the quarter.
05 - numeric

FL05 HS_DIU_OF_TRIM HS_DIU_OF_QUA
Hours flown during the day as other function on board in the quarter.
05 - numeric

FL06 HS_NOT_1P_TRIM HS_NOC_1P_QUA
Hours flown during the night as first pilot in the quarter.
05 - numeric

FL07 HS_NOT_2P_TRIM HS_NOC_2P_QUA
Hours flown during the night as second pilot in the quarter.
05 - numeric

FL08 HS_NOT_OF_TRIM HS_NOC_OF_QUA
Hours flown during the night as other function on board in the quarter.
05 - numeric

FL09 NUM_CART_IFR NUM_IFR_CARD
Number of the IFR card.
08 - numeric

FL10 OM_CART_IFR UNIT_IFR
Military organization that emitted the IFR card.
03 - numeric (code)

FL11 DAT_VENC_IFR DT_EXP_IFR_CARD
Date of expiration of the IFR card.
06 - numeric

FL12 OM-CART_SAU UNIT_HEALTH
Military organization that emitted the health card.
03 - numeric (code)

FL13 DAT_VENC_SAU DT_EXP_HEALTH
Date of expiration of the health card (dd/mm/yy).
06 - numeric

FL14 POSTO_VOO RANK_FLIGHT
Rank of the military during the flight.
02 - alphanumeric (code)

CO01 TIPO_CURSO TYPE_COURSE
Type of course according to existing code.
06 - alphanumeric (code)

CO02 DAT_IN_CUR DT_START_COURSE
Date of start of the course (dd/mm/yy).
06 - numeric

CO03 DAT_TERM_CUR DT_END_COURSE
Date of end of the course (dd/mm/yy).
06 - numeric

CO04 MEDIA_CUR GRADE_COURSE
Average grade during the course.
04 - alphanumeric

CO05 CLASSIF_CUR CLASSIF_COURSE
Classification in the course (20 of 50).
07 - alphanumeric

CO06 AREA_CUR AREA_COURSE
Area of interest of the course according to existing code.
02 - numeric (code)

CO07 NOME_CUR NAME_COURSE
Name of the course.
40 - alphanumeric

CO08 NIVEL_CUR LEVEL_COURSE
Level of the course according to existing code.
02 - alphanumeric (code)

CO09 TERMINO_CUR FIN_COURSE
Flag informing that finished this course.
01 - alphabetic

ME01 TIPO_MEDAL TYPE_MEDAL
Type of medal according to existing code.
02 - numeric (code)

ME02 DAT_MEDAL DT_MEDAL
Date of receiving the medal.
06 - numeric

ME03 DAT_DECENIO DT_DECENNIUM
Date of completing the decennium.
06 - numeric

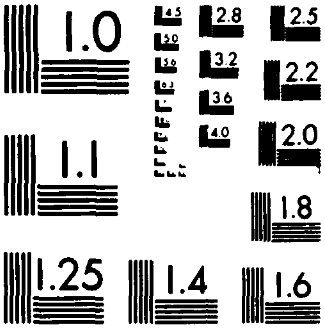
ME04 GRAU GRADE_MEDAL
Grade of the medal according to existing code.
01 - numeric (code)

ME05 RES_MEDAL SUM_MEDAL
Summary of information about the medal.
30 - alphanumeric

QU01 QUADRO_ESP ACTIVE_LIST

AD-A182 692   A DATABASE DESIGN FOR THE BRAZILIAN AIR FORCE MILITARY   2/2
PERSONNEL CONTROL SYSTEM(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI..   W MUSSATO
UNCLASSIFIED   JUN 87 AFIT/GCS/ENG/87J-1   F/G 5/9   NL

END
8-87
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Active list of the military.
02 - numeric (code)

QU02 MOT_AGREG RES_NON_DUTY_STA
Reason for non duty status.
02 - numeric (code)

QU03 DAT_AGREG DT_NON_DUTY_STA
Date of start the non duty status.
06 - numeric

QU04 DAT_REVERS DT_RETURN
Date of returning to duty status.
06 - numeric

QU05 DAT_INCL DT_INCL_ACT_LIST
Date of inclusion on the active list.
06 - numeric

QU06 DAT_INCL_EXT DT_INCL_EXT_NUM
Date of inclusion on the extra numeric situation.
06 - numeric

QU07 DAT_EXCL_EXT DT_EXCL_EXT_NUM
Date of exclusion on the extra numeric situation.
06 - numeric

ES01 DAT_INIC_PRTSV DT_START_EXT_LOS
Date of start extension of length of service.
06 - numeric

ES02 DAT_TERM_PRTSV DT_END_EXT_LOS
Date of end extension of length of service.
06 - numeric

ES03 DAT_ESTABIL DT_STABIL
Date of acquiring stability.
06 - numeric

ES04 NUM_ESTABIL NUM_STABIL
Number of times extended length of service.
01 - numeric

EX01 NAT_EXCLUS REA_EXCL
Reason for the exclusion of active duty.
02 - numeric (code)

EX02 DAT_EXCLUS DT_EXCL
Date of exclusion of the active duty.
06 - numeric

EX03 NAT_FALEC CAUSE_DEATH
Cause of death (during service or not).

01 - numeric (code)

EX04 RES_EXCL SUM_EXCL
Summary of the reason for exclusion of the active duty.
30 - alphanumeric

UN01 ABREV ABREV
Abreviation of the name of the Military Organization (Unit).
06 - alphanumeric

UN02 NOME NAME
Full Name of the Unit.
40 - alphanumeric

UN03 PREV_POSTO_QUADRO PRED_RANK_SPE
A prediction of the number of military within the Rank and Specialty.
04 - numeric

UN04 EFF_POSTO_QUADRO EFF_RANK_SPE
An effective number of military within the Rank and Specialty.
04 - numeric

UN05 LOCAL LOCAL
Localization of the Unit, normally city or part of the city.
02 - numeric (code)

UN06 COM_REG REG_COM
Regional Command where the Unit is located (1-7).
01 - numeric

UN07 GDE_COM MAJCOM
A Major Command of subordination.
07 - alphanumeric

Appendix B

Description of Entities and Relationships

This Appendix presents a brief description of the entities, weak entities and relationships identified in Chapter III.

Along with the description are presented also their attributes. Attributes preceded by a *(star) are differentiated from others because they are identifiers of the entity, weak entity and relationship.

Entities

Personnel - *rec_num, name, state_birth, dt_birth, father_name, mother_name, id_num, sex, soc_sec_num, inc_tax_num, med_rec_num, tag_name, dt_exp_health, unit_health, dt_stabil, dt_decennium.

Personnel - the key attribute is rec_num which is the number that each person receives when entering the Air Force. This entity contains the personnel information related to the person, such as name, state of birth, date of birth, father name, mother name, identification number, sex, social security number, income tax number, expiration date of the health card, unit that issued the health card, date of acquired stability, date of decennium.

Aviator - *rec_num, tag_name.

Aviator - aviator is a personnel. Aviator is a specialization of personnel that is related to flight and ifr_card, and personnel is not. Aviator inherits the attributes of personnel and has the same key attribute. Tag_name is a unique nonkey attribute to be used by aviator.

Leaving - *type_leave, name_leave.

Leaving - the key attribute is type_leave which is a code for each type of leave. Name_leave is the full name of the leave.

Promotion - *rank.

Promotion - the key attribute is the rank and also the unique attribute.

Course - *type_course, area_course, level_course, name_course.

Course - the key attribute is type_course which is the code of the course. Name_course is the full name of the course, area_course is the related area for the course, such as, electronics, logistics, etc. Level_course is the status, graduate, person specialization, etc.

Medal - *type_medal, grade_medal, sum_medal.

Medal - the key attribute is type_medal which is a codification of the medal, name_medal is the full name of the medal, and grade_medal is a hierarchical graduation for the medal.

Exclusion - *rea_excl, sum_excl.

Exclusion - the key attribute is rea_excl which is the code to reason for exclusion, sum_excl is the summary of the reason for the exclusion. Exclusion is an entity because can occur more than once, i.e., the person can be excluded, included and excluded again.

Death - *rea_excl, sum_excl.

Death - death is a exclusion, a special kind of exclusion can be death, because has the attribute cause_death in the relationship with personnel, in this case, has the same attributes as exclusion.

Inclusion - *rea_incl, sum_incl.

Inclusion - the key attribute is rea_incl which is the code to reason for inclusion, sum_incl is the summary of the reason for the inclusion. Similar to exclusion, the person can be included more than once in the Air Force, keeping the same rec_num.

Specialty - *active_list.

Specialty - the key attribute is active_list (specialty) which means engineers, aviators, medical_ doctors, etc. This is the unique attribute for the entity.

Ifr_card - *num_ifr_card, dt_exp_ifr_card, unit_ifr.

Ifr_card - the key attribute is num_ifr_card which is the number associated to each ifr_card. Dt_exp_ifr_card and unit_ifr, which is the unit that issued the card, are the other attributes. This card is only issued for pilots, i.e., entity aviator.

Unit - *abbrev, name_unit, local, reg_com, majcom.

Unit - the key attribute is abbrev which is a code for each existing unit, name_unit is the full name of the unit, local, regional command (reg_com), and major command (maj_com), are the others attributes of the entity.

Weak Entities

Stability - *num_stabil, dt_start_ext_los, dt_end_ext_los.

Stability - the discriminator attribute is num_stabil, which is the number of times the person has his length of service extended until acquiring stability. Dt_start_ext_los and dt_end_ext_los are attributes to indicate start and end of each extension of length of service.

Flight - *year_ref, *qua_ref, hs_1p_diu_qua, hs_2p_diu_qua, hs_of_diu_qua, hs_1p_noc_qua, hs_2p_noc_qua, hs_of_noc_qua.

Flight - the discriminator attributes are year_ref and qua_ref, since all information about flight is stored within year and quarter. Hours flew as 1p (first pilot), 2p (second pilot) and of (other function), are recorded during diurnal or nocturnal operations.

Ext_num - *num_ext_num, dt_incl_ext_num, dt_excl_ext_num.

Ext_num - the discriminator attribute is num_ext_num, which is the number of times the person was included in such case, i.e., is not allowed to fly for some health problem. Dt_incl_ext_num and dt_excl_ext_num are dates of inclusion and exclusion of the person as extra numerical, i.e., has no number in the promotion list. This is an weak entity because it is dependent on aviator.

Moving - *dt_moving, dt_pres, dt_detach, unit_mov, sit_unit, sum_mov.

Moving - the discriminator attribute is dt_moving, which is the moving date for the person to some unit. Dt-pres and dt_detach are arriving and leaving date of this unit. Unit_mov is the unit from where came the person. Sit_unit is the status of the person in the unit, instructor, commandant, etc. Sum_mov is the summary of the moving. Moving is a weak entity that depends on personnel and unit.

Nomination - *dt_nom, dt_exo, sit_unit_nom, sum_nom.

Nomination - the discriminator attribute is dt_nom, which is the date of nomination. Dt_exo is the date of exoneration of some duty. Sit_unit_nom is the status of the person in the unit, commandant, assistant dean, etc. Sum_nom is the summary of the nomination. Nomination is a weak entity that depends on personnel and unit.

Designation - *dt_desig, dt_waiver, sit_unit_des, sum_des.

Designation - the discriminator attribute is dt_desig, which is the date of designation. Dt_waiver is the date of waiver of some duty. Sit_unit_des is the status of the person in the unit, instructor, assistant dean, etc. Sum_des is the summary of the designation. Designation is a weak entity that depends on personnel and unit.

Attachment - *dt_start_att, dt_end_att, rea_att, sit_unit_att, sum_att.

Attachment - the discriminator attribute is dt_start_att, which is the starting date of attachment. Dt_end_att is the ending date of attachment. Sit_unit_att is the status of the person in the unit, instructor, assistant dean, etc. Sum_att is the summary of the attachment. Rea_att is a codification of the reason for the attachment. Attachment is a weak entity that depends on personnel and unit.

Non_duty - *dt_non_duty_sta, dt_return, sum_non_duty_sta.

Non_duty - the discriminator attribute is dt_non_duty_sta, which means the date on which the person became non duty status. Dt_return is the returning date to duty status, and sum_non_duty_sta is the the summary of the reason for non duty status. Non_duty is a weak entity that depends on personnel and specialty. It is also defined as an entity because of the historical data.

Relationships

Personnel_leave (per_lea) - dt_start_leave, dt_end_leave.

Per_lea - this is a many-to-many relationship since each person can take more than one leave, although not at the same time, and this historical data is maintained for future reference. A type of leave can be related to more than one person.

Personnel_promotion (per_pro) - dt_pro, crit_pro.

Per_pro - this is a many-to-many relationship since each person, during his career, can have more than one promotion, and any type of rank can be related to more than one person. Dt_pro is the data of promotion and crit_pro is the criteria for promotion, merit or antiquity.

Personnel_course (per_cou) - dt_start_course, dt_end_course, grade_course, classif_course, fin_course.

Per_cou - this is a many-to-many relationship since each person can take more than one course, and any type of course can be related to more than one person. Dt_start_course is the starting date of the course, and dt_end_course is the predicted date for the end of the course, fin_course is the indication that the course finished. Grade_course is the GPA for the course, and classif_course is the relative position in the course, ex: 34 of 123.

Personnel_medal (per_med) - dt_medal.

Per_med - this is a many-to-many relationship since each person can have more than one medal, and each type of medal can be received by more than one person. Dt_medal is the receiving date of the medal.

Personnel_exclusion (per_exc) - dt_excl.

Per_exc - this is a many-to-many relationship since, as explained before, each person can have more than one exclusion, and each type of exclusion can be related to more than one person. Dt_excl is the date of exclusion of the person.

Personnel_death (per_dea) - cause_death.

Per_dea - this is a one-to-one relationship where death is a special kind of exclusion. It is kept separated because of the attribute cause_death, which is the information about the cause of death i.e., if the person died in service (duty hours) or not.

Personnel_inclusion (per_inc) - dt_incl.

Per_inc - this is a many-to-many relationship since, as explained before, each person can have more than one inclusion, and each type of inclusion can be related to more than one person. Dt_incl is the date of inclusion of the person.

Personnel_specialty (per_spe) - dt_incl_act_list.

Per_spe - this is a many-to-many relationship since each person can be assigned to more than one specialty, i.e., can change the previous specialty, and each specialty can be related to more than one person. Dt_incl_act_list is the date of inclusion in the new active list, i.e., date of change the specialty.

Personnel_unit (per_unit) - no attributes.

Per_unit - this is a many-to-one relationship that shows the current unit of the person, each unit can be related to more than one person, but each person is assigned to only one current unit.

Aviator_promotion_flight (avi_pro_fli) - no attributes.

Avi_pro_fli - this is a many-to-many-to-many relationship where each aviator can be related to more than one rank and more than one flight. This occurs when dealing with historical data.

Aviator_ifr_card (avi_ifr) - no attributes.

Avi_ifr - this is a one-to-one relationship where for each aviator there exists one ifr_card, and each ifr_card is assigned to one aviator.

Aviator_ext_num (avi_ext) - no attributes.

Avi_ext - this is a one-to-many relationship where each aviator can stay in the situation of extra numerical more than once.

Personnel_stability (per_sta) - no attributes.

Per_sta - this is a one-to-many relationship where for each person there exists more than one stability.

Personnel_specialty_non_duty (per_spe_non) - no attributes.

Per_spe_non - this is a many-to-many-to-many relationship where each person related to specialty can have more than one non duty status, considering historical data.

Personnel_unit_moving (per_unit_mov) - no attributes.

Per_unit_mov - this is a many-to-many-to-many relationship where each person can be related to more than one unit, considering historical data, and each unit can be related to more than one person. Is also possible to have more than one moving to the same per_unit.

Personnel_unit_nomination (per_unit_nom) - no attributes.

Per_unit_nom - this is a many-to-many-to-many relationship where each person can be related to more than one unit, considering historical data, and each unit can be related to more than one person. Is also possible to have more than one nomination to the same per_unit.

Personnel_unit_designation (per_unit_des) - no attributes.

Per_unit_des - this is a many-to-many-to-many relationship where each person can be related to more than one unit, considering historical data, and each unit can be related to more than one person. Is also possible to have more than one designation to the same per_unit.

Personnel_unit_attachment (per_unit_att) - no attributes.

Per_unit_att - this is a many-to-many-to-many relationship where each person can be related to more than one unit, considering historical data, and each unit can be related to more than one person. Is also possible to have more than one attachment to the same per_unit.

Unit_promotion_specialty (unit_pro_spe) - pred_rank_spe, eff_rank_spe.

Unit_pro_spe - this is a many-to-many-to-many relationship where, each unit needs to store informations about the predicted and effective number of persons, within each rank and specialty.

Appendix C

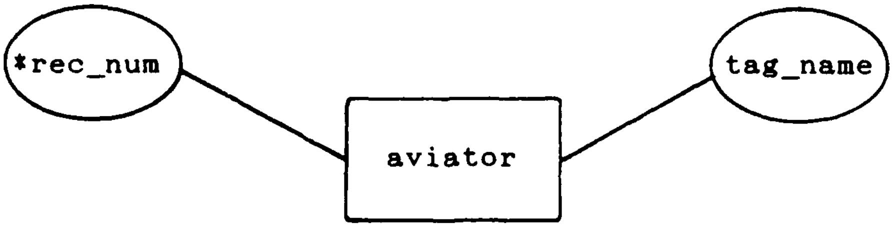
ER Diagram for the Personnel Headquarters

In this Appendix, a complete ER Diagram shows where the defined attributes are related to entities, weak entities and relationships.

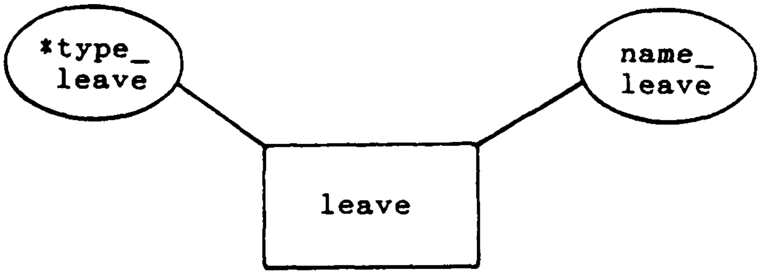
The *(star) symbol means that such attribute(s) uniquelly identify the entity or relationship. In the case of weak entity, the star symbol means attributes that can uniquelly identify the entity when connected to another entity.

At the end of this Appendix is also presented the ER Diagram that reflects only the relationships among the identified entities.

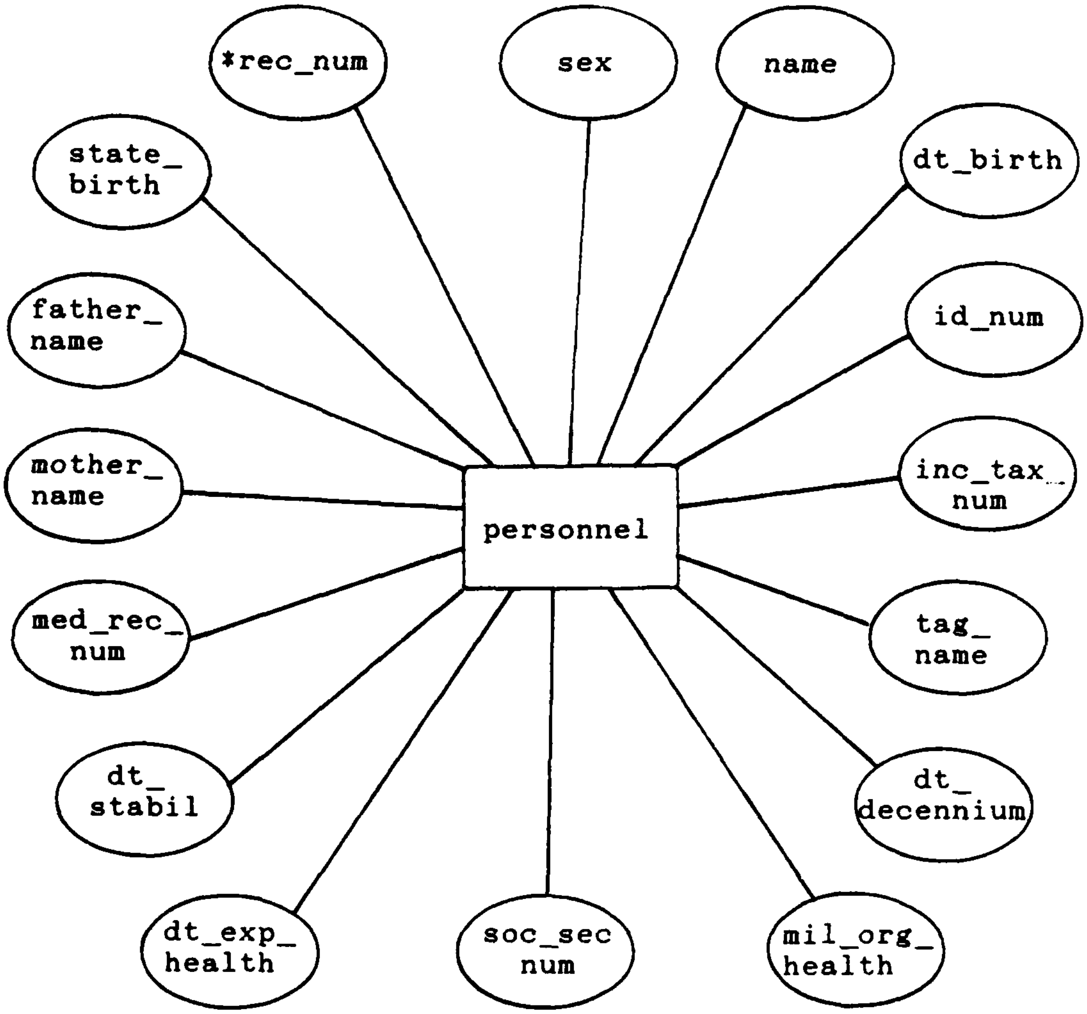
1. Attributes of entities.



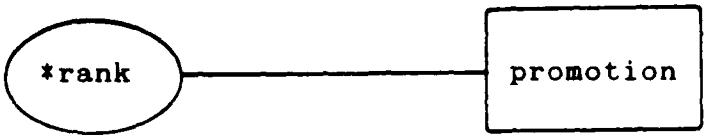
Entity aviator



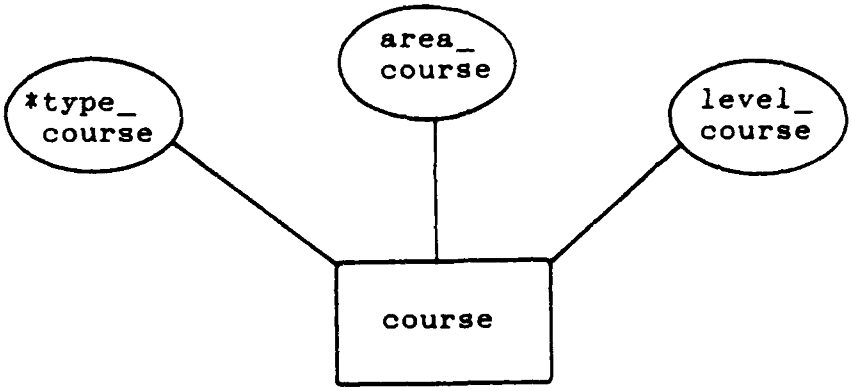
Entity leave



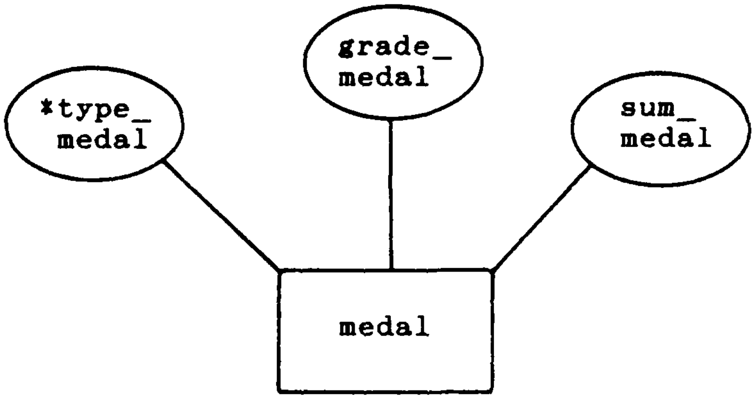
Entity personnel



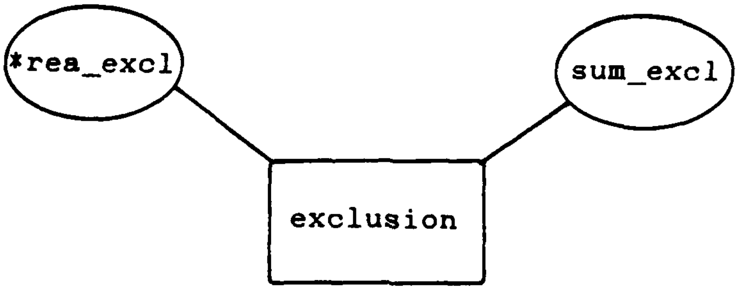
Entity promotion



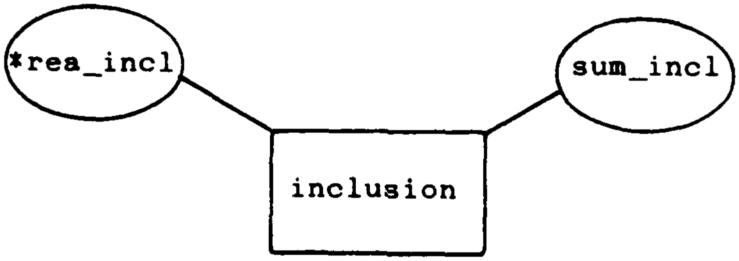
Entity course



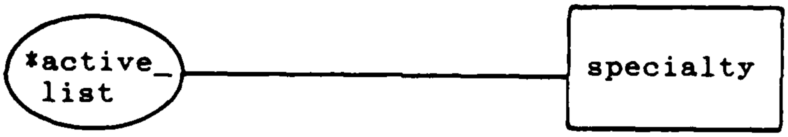
Entity medal



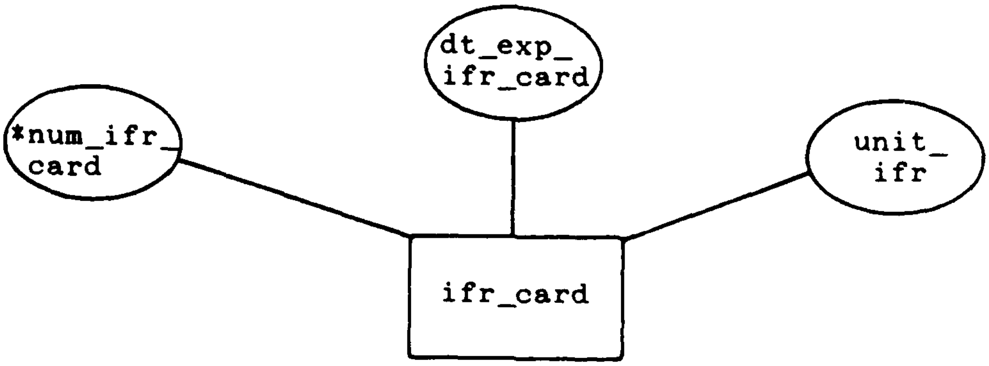
Entity exclusion



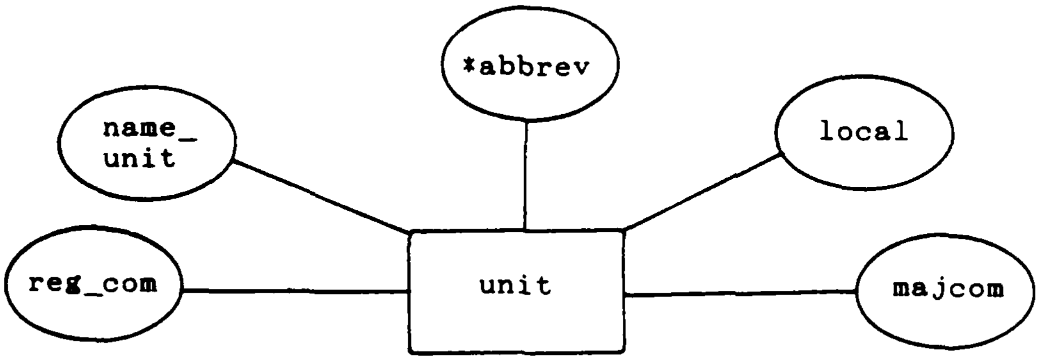
Entity inclusion



Entity specialty

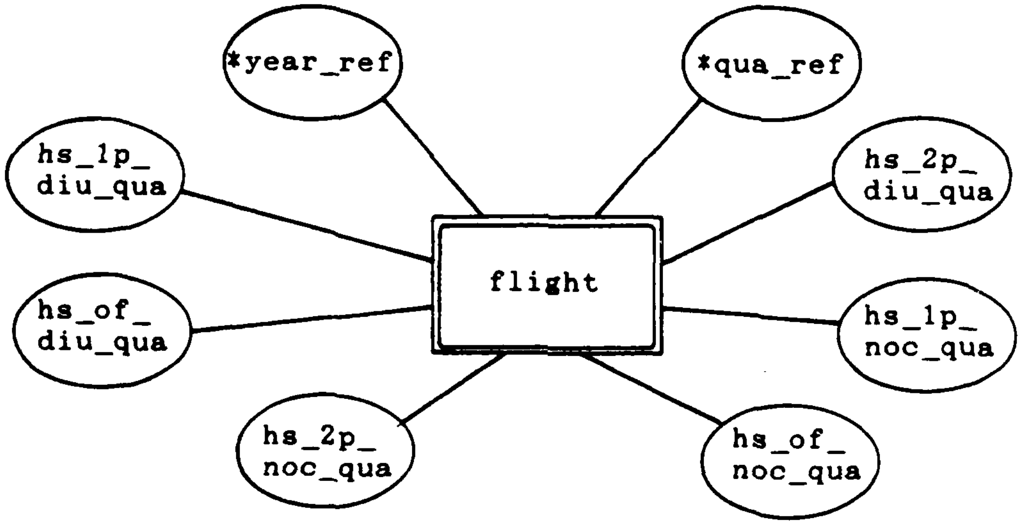


Entity ifr_card

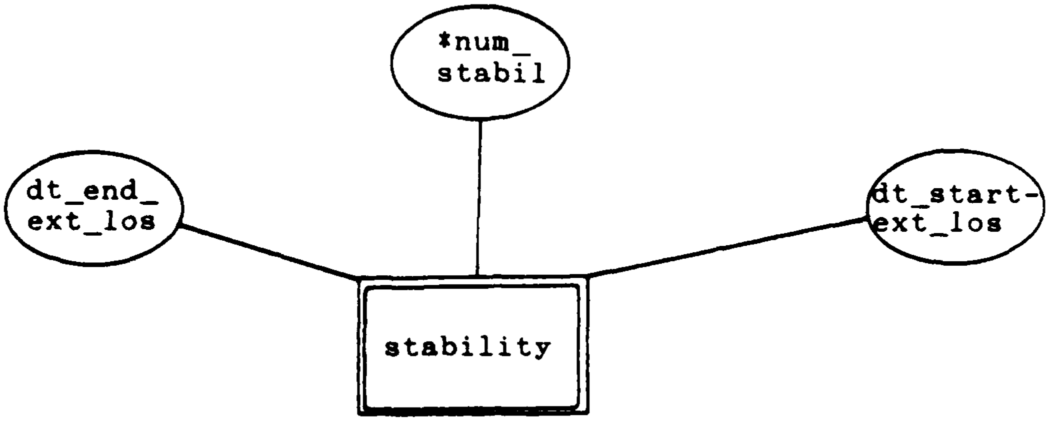


Entity unit

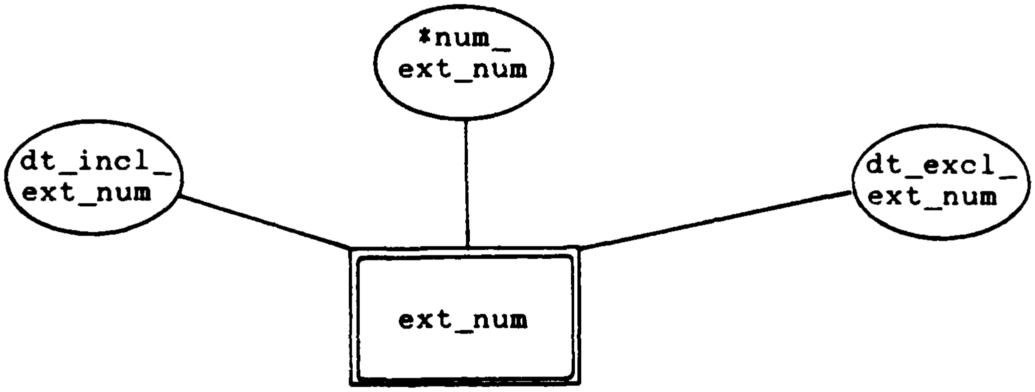
2. Attributes of weak entities.



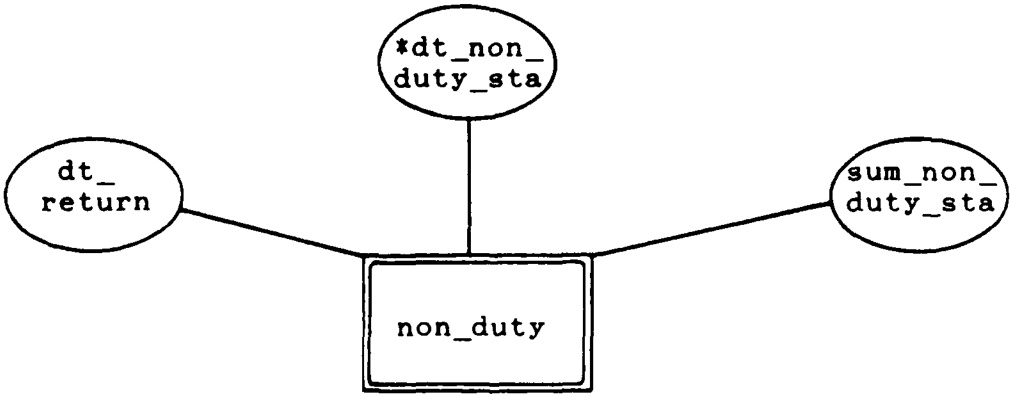
Weak entity flight



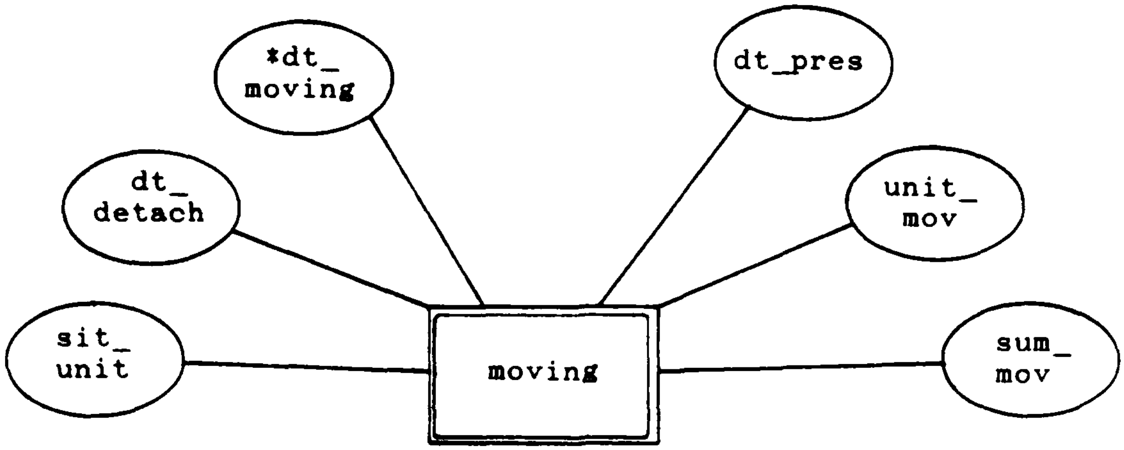
Weak entity stability



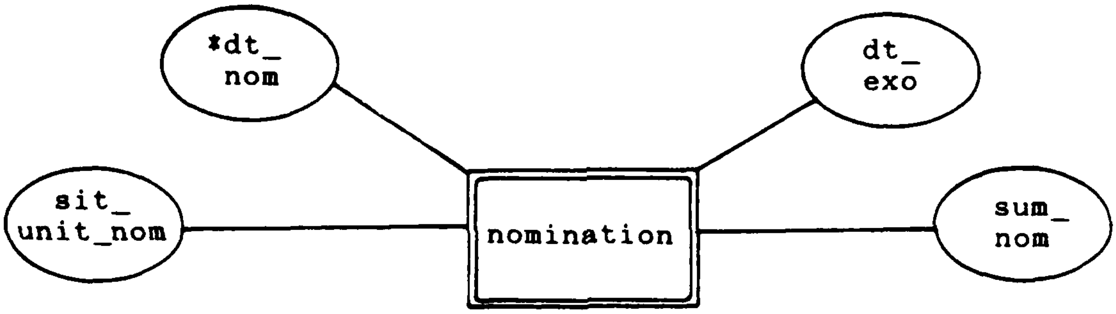
Weak entity ext_num



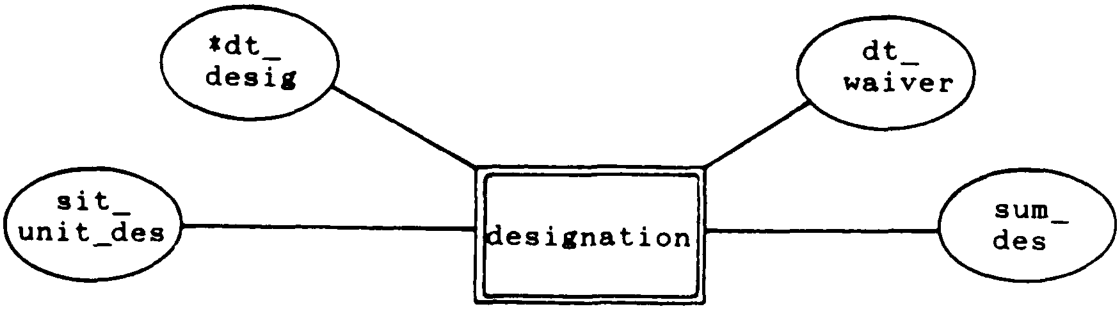
Weak entity non_duty



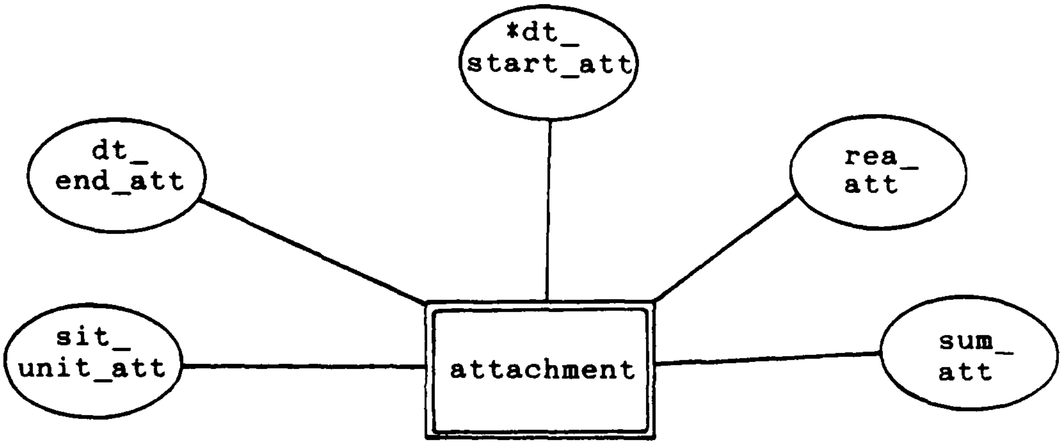
Weak entity moving



Weak entity nomination

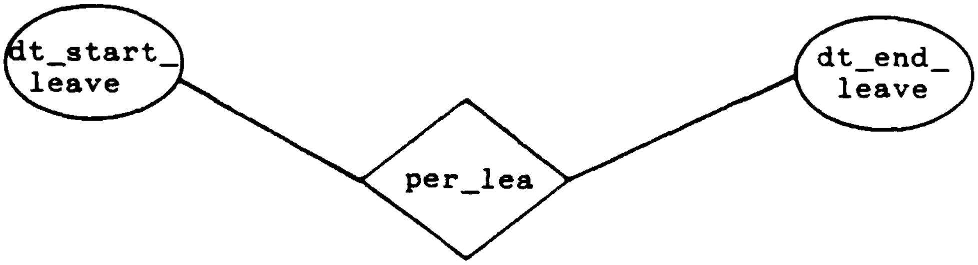


Weak entity designation

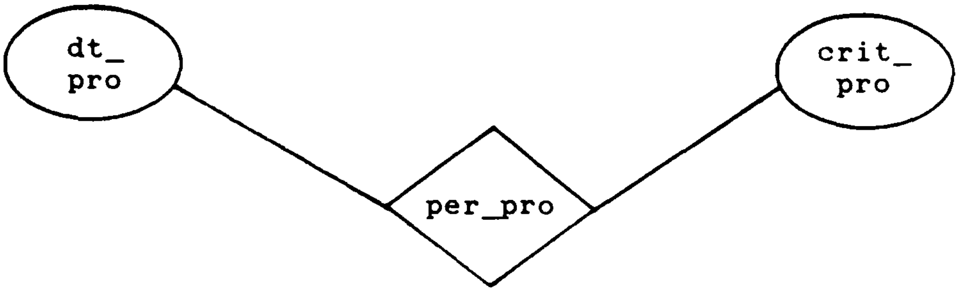


Weak entity attachment

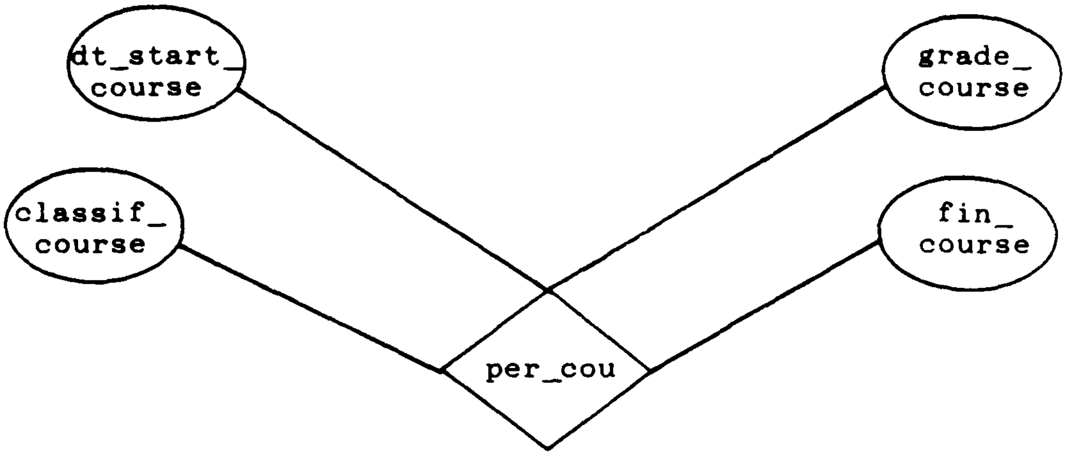
3. Attributes of relationships.



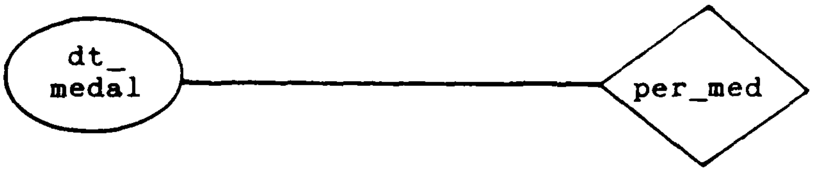
Relationship personnel_leave



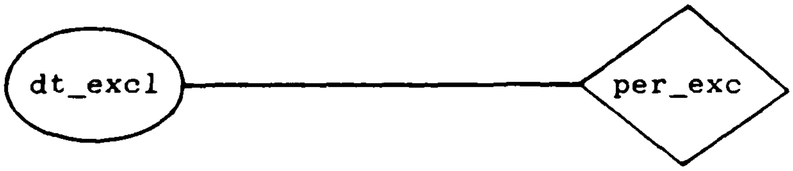
Relationship personnel_promotion



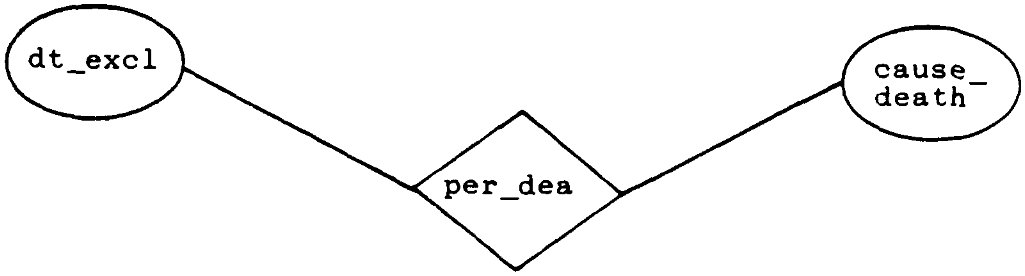
Relationship personnel_course



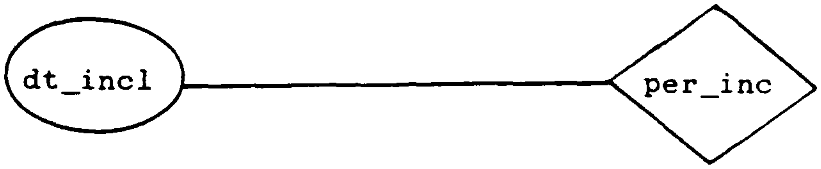
Relationship personnel_medal



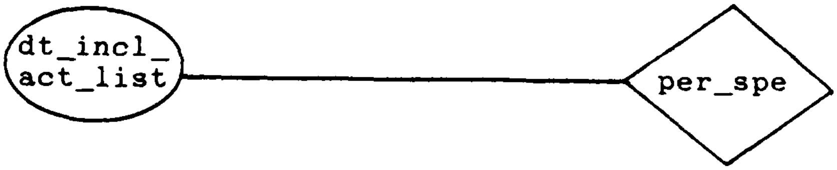
Relationship personnel_exclusion



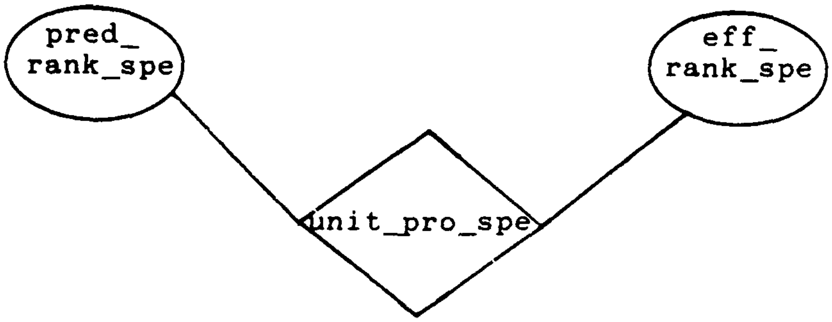
Relationship personnel_death



Relationship personnel_inclusion

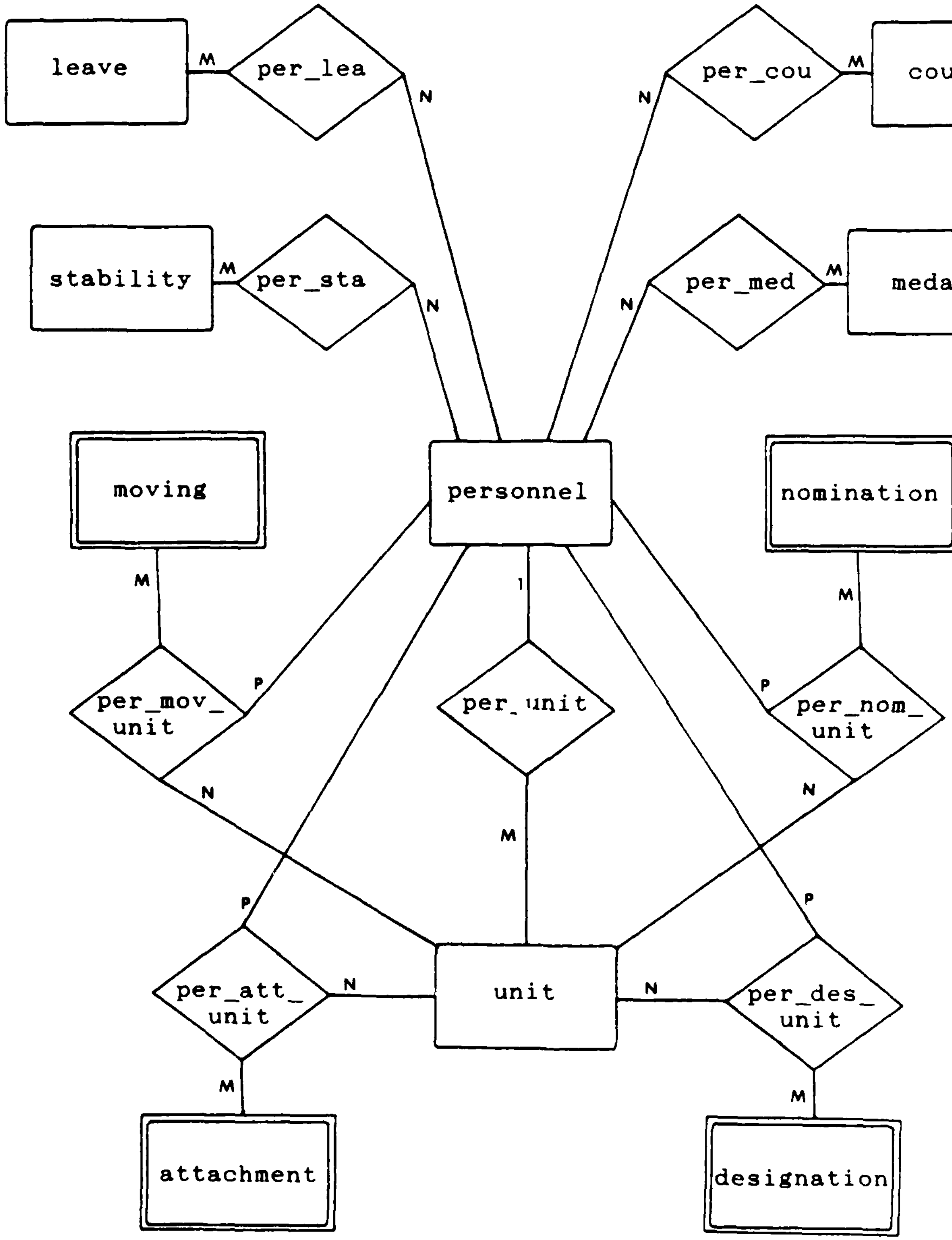


Relationship personnel_specialty

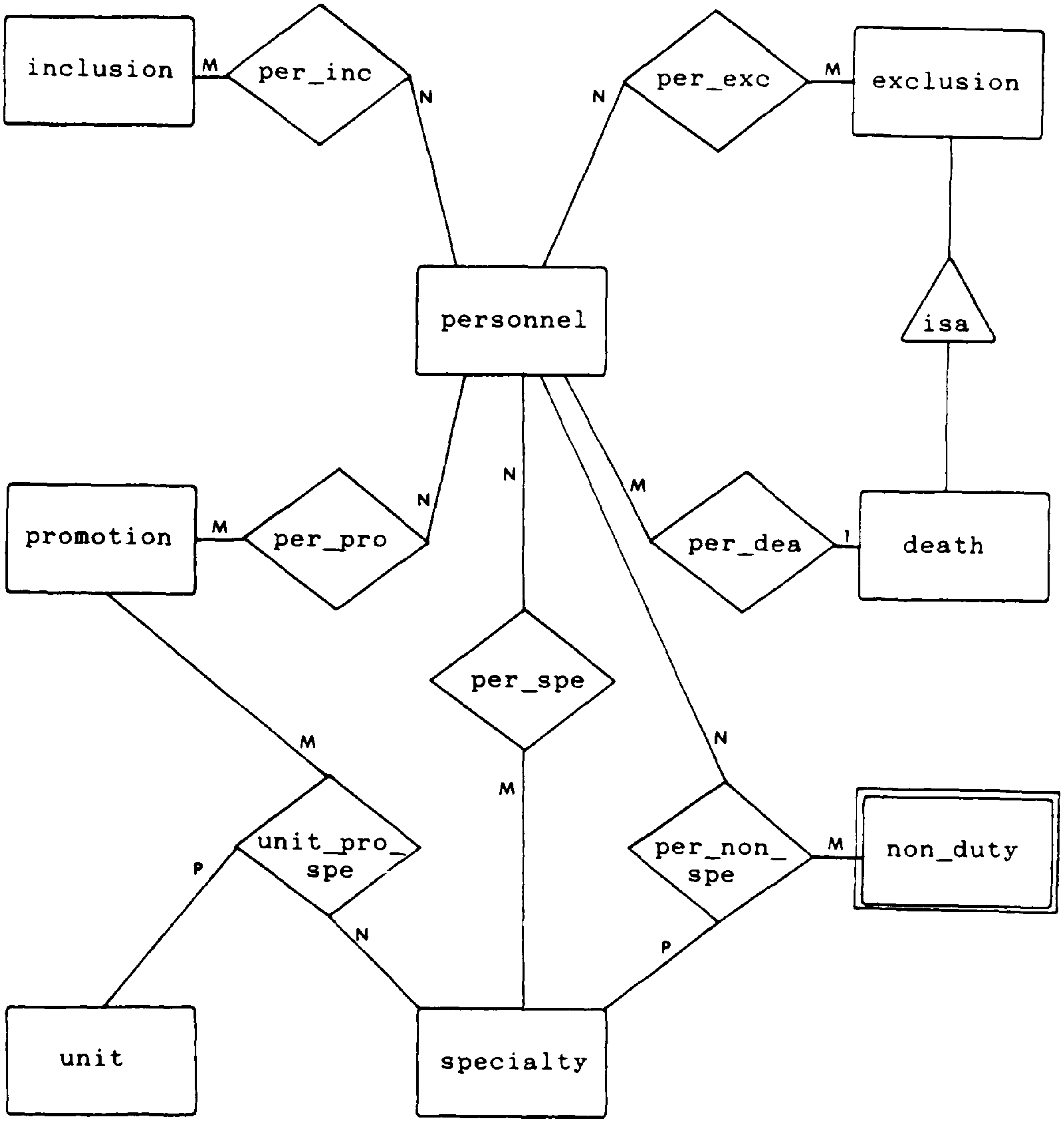


Relationship unit_promotion_specialty

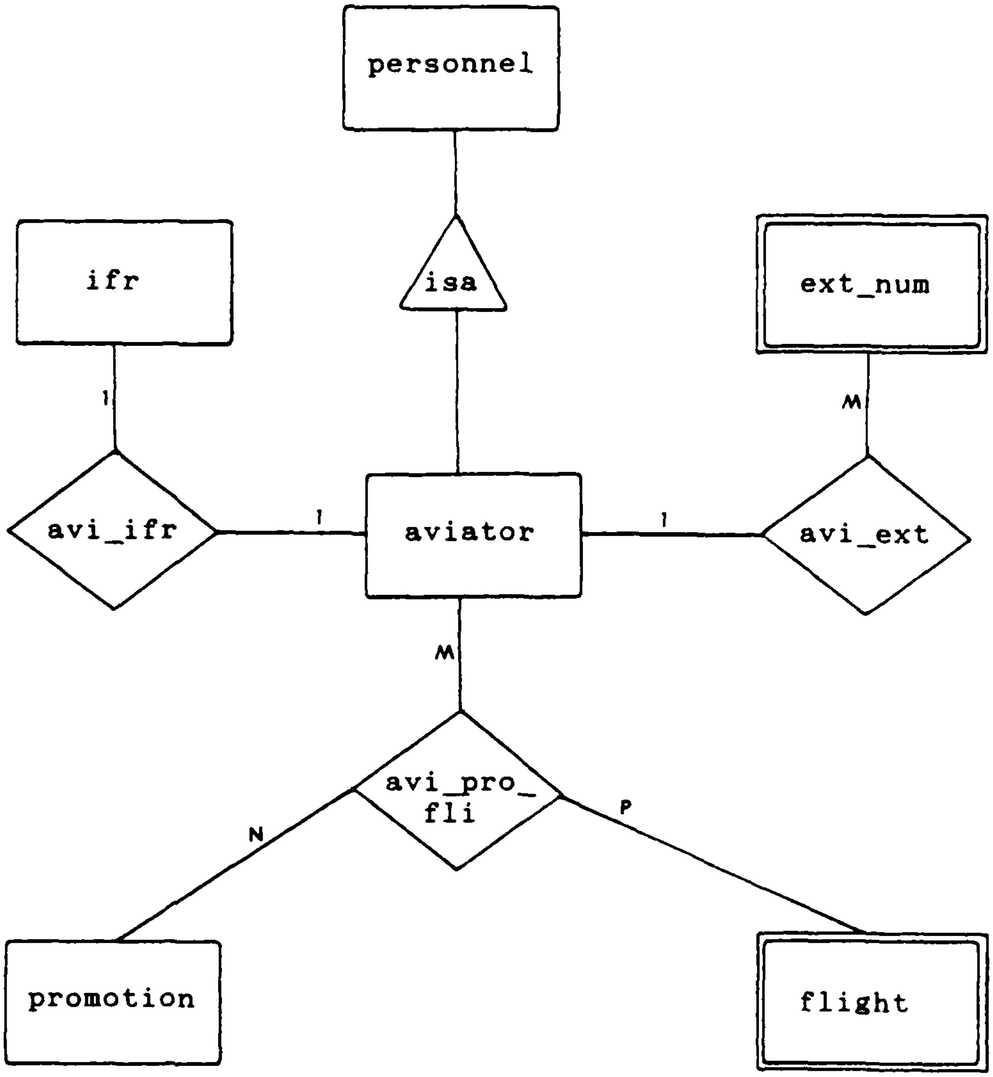
4. ER Diagram.



Personnel headquarters' ER Diagram.



Personnel headquarters' ER Diagram.



Personnel headquarters' ER Diagram.

Appendix D

Analysis of the relations until BCNF

In Chapter IV, the transition from ER Diagram to Relations was explained, in the same chapter some analysis were done to show the normalization process until Boyce/ Codd Normal Form (BCNF).

This Appendix contains the transition process from the ER Diagram, shown at the end of Chapter III, to the relations used in Chapter IV. This Appendix contains also, a detailed analysis of the normalization process of those relations, until BCNF.

Entity --> Relation

As explained in Chapter IV, the process used to transform an entity into a relation is to get the key attribute from the entity and make it primary key of the relation, and the nonkey attributes of an entity now become nonkey attributes of the relation.

1 - Personnel

```
rec_num --> name, state_birth, dt_birth, father_name,
            mother_name, id_num, sex, soc_sec_num,
            inc_tax_num, med_rec_num, tag_name,
            dt_exp_health, unit_health, dt_stabil
```

Rec_num is the primary key for the relation personnel, it was selected for being the first number received by the military, when included in the Air Force, and stay with him during his entire active duty.

Some attributes were considered as potential candidate key for the relation, but each one has a problem to meet the requirements to be the candidate key.

Id_num could be a candidate key, but, the problem is that in case of airmen, they receive the rec_num as soon as they are included in the Air Force, and only a few weeks later they receive their id_num. In fact, the identification process is now being changed, in order to use the same rec_num as id_num, this procedure shows that in no longer future the id_num will be replaced by rec_num, becoming the only identification number for the military.

Soc_sec_num is only used for the military within the Air Force, in the case of the Airmen, that stay in the Air Force for at most two years, they do not have such number.

Inc_tax_num is only given to a military, after the presentation of the income tax form, and it occurs normally a few months after the military has being included in the Air Force.

Med_rec_num is given to a military during his first visit to an hospital, in this case may take a long time to get such number. The health headquarters is changing this procedure in order to use the rec_num as med_rec_num, so, in a brief future this number will not be necessary, following the same procedure as id_num.

Name, state_birth, and dt_birth could be considered candidate key, but no one can be sure that those three attributes can not be appear more than once.

Name, father_name, and mother_name could also be considered candidate key for the relation, but, no one can guarantee their to be unique.

No other attribute could be considered candidate key for the relation.

The relation personnel is in 3NF, since each attribute has an atomic value, every nonkey attribute is fully dependent on rec_num, and also every nonkey attribute is nontransitively dependent on rec_num.

The relation personnel is in BCNF since the unique existent determinant is the primary key.

2 - Aviator

rec_num --> tag_name

The relation aviator is a specialization of personnel, and use the same primary key, rec_num. Tag_name is the unique nonkey attribute of the relation. The relation is in 3NF and BCNF for the reasons presented in the relation personnel.

3 - Leave

type_leave --> name_leave
name_leave --> type_leave

Type_leave is the primary key for the relation leave, because can uniquely identify the relation.

Name_leave is a candidate key for the relation, but, since type_leave is a codification of name_leave, was selected to be the primary key, because is shorter than name_leave, and expected to have less typing errors than name_leave.

The relation leave is in 3NF, since each attribute has an atomic value, every nonkey attribute is fully dependent on type_leave and also every nonkey attribute is nontransitively dependent on type_leave.

The relation leave is in BCNF since the existent determinants, type_leave and name_leave are candidate keys.

4 - Promotion

Rank is the primary key and unique attribute of the relation. It is clear to see that the relation is in 3NF and BCNF.

5 - Course

```
type_course --> area_course, level_course, name_course
name_course --> area_course, level_course, type_course
```

Type_course is the primary key for the relation, because can uniquely identify the relation.

Name_course is a candidate key for the relation, but, as explained in the relation leave, since type_course is a codification name_course, it was selected to be the primary key for the relation.

The relation course is in 3NF, since each attribute has an atomic value, every nonkey attribute is fully dependent on type_course and also every nonkey attribute is nontransitively dependent on type_course.

The relation course is in BCNF since the existent determinants, type_course and name_course are candidate keys.

6 - Medal

```
type_medal --> grade_medal, sum_medal
sum_medal  --> grade_medal, type_medal
```

Type_medal is the primary key for the relation medal, because can uniquely identify the relation.

Sum_medal is a candidate key for the relation, but, is included in the same situation as type_course and name_course, so, type_medal was selected to be the primary key for the same arguments shown before.

The relation medal is in 3NF, since each attribute has an a⁺omic value, every nonkey attribute is fully dependent on type_medal and also every nonkey attribute is nontransitively dependent on type_medal.

The relation medal is in BCNF since the existent determinants, type_medal and sum_medal are candidate keys.

7 - Exclusion

```
rea-excl --> sum_excl
sum_excl --> rea_excl
```

Rea_excl is the primary key for the relation exclusion, because can uniquely identify the relation.

Sum_excl is a candidate key for the relation, but, since was included in the same case as type_leave and name_leave, rea_excl was selected to be the primary key for the same reasons explained before.

The relation exclusion is in 3NF, since each attribute has an atomic value, every nonkey attribute is fully dependent on rea_excl and also every nonkey attribute is nontransitively dependent on rea_excl.

The relation exclusion is in BCNF since the existent determinants, rea_excl and sum_excl are candidate keys.

8 - Death

Relation death has the same primary key and attributes as exclusion, plus the attribute cause_death. All observations used for exclusion are applicable in death.

9 - Inclusion

```
rea-incl --> sum_incl
sum_incl --> rea_incl
```

Rea_incl is the primary key for the relation inclusion, because can uniquely identify the relation.

Sum_incl is a candidate key for the relation, but, since was included in the same case as type_leave and name_leave, rea_excl was selected to be the primary key for the same reasons explained before.

The relation inclusion is in 3NF, since each attribute has an atomic value, every nonkey attribute is fully dependent on rea_incl and also every nonkey attribute is nontransitively dependent on rea_incl.

The relation inclusion is in BCNF since the existent determinants, rea_incl and sum_incl are candidate keys.

10 - Specialty

Active_list is the primary key and unique attribute of the relation specialty. The relation is clearly in 3NF and BCNF.

11 - Ifr_card

num_ifr_card --> dt_exp_ifr_card, unit_ifr

Num_ifr_card is the primary key for the relation, and no other attribute can be considered as candidate key.

The relation ifr_card is in 3NF, since each attribute has an atomic value, every nonkey attribute is fully dependent on num_ifr_card and also every nonkey attribute is nontransitively dependent on num_ifr_card.

The relation ifr_card is in BCNF since the unique existent determinant num_ifr_card is the primary key.

12 - Unit

abbrev --> name_unit, local, reg_com, majcom
name_unit --> local, reg_com, majcom, abbrev

Abbrev is the primary key for the relation unit, because can uniquely identify the relation.

Name_unit is a candidate key for the relation, but, is in the same situation as type_course and name_course, the attribute abbrev was selected to be the primary key for the same arguments presented before.

The relation unit is in 3NF, since each attribute has an atomic value, every nonkey attribute is fully dependent on abbrev and also every nonkey attribute is nontransitively dependent on abbrev.

The relation unit is in BCNF since the existent determinants, abbrev and name_unit are candidate keys.

Weak Entity --> Relation

As explained in chapter IV, the process to transform a weak entity into a relation, is to get the discriminator from the weak entity, and the key attribute from the relation on which the weak entity is dependent, to form the primary key for the relation, and the nonkey attributes of the weak entity will become nonkey attributes of the relation.

13 - Stability

Stability is dependent on personnel, which key attribute is rec_num.

```
rec_num, num_stabil --> dt_start_ext_los,
                        dt_end_ext_los

rec_num, dt_start_ext_los --> num_stabil,
                              dt_end_ext_los
```

The primary key for the relation stability is composed of rec_num and num_stabil, because they can uniquely identify the relation. Rec_num and dt_start_ext_los are candidate keys for the relation, num_stabil was selected to be the primary key, because has a value shorter than dt_start_ext_los.

The relation stability is in 3NF, since each attribute has an atomic value, every nonkey attribute is fully dependent on rec_num and num_stabil, and also every nonkey attribute is nontransitivelly dependent on rec_num and num_stabil.

The relation stability is in BCNF since the existent determinants, (rec_num, num_stabil) and (rec_num, dt_start_ext_los) are candidate keys.

14 - Flight

Flight is dependent on aviator and promotion, which key attribute are rec_num and rank.

```
rec_num,   ¦        hs_1p_diu_qua, hs_2p_diu_qua,
rank,      ¦-->     hs_of_diu_qua, hs_1p_noc_qua,
year_ref,  ¦        hs_2p_noc_qua, hs_of_noc_qua
qua_ref    ¦
```

The primary key for the relation flight is composed of rec_num, rank, year_ref, qua_ref, because they can uniquely identify the relation, no other attribute can be used as a candidate key.

The relation flight is in 3NF, since each attribute has an atomic value, every nonkey attribute is fully dependent on rec_num, rank, year_ref, and qua_ref, also every nonkey attribute is nontransitively dependent on rec_num, rank, year_ref, and qua_ref.

The possible dependence year_ref and qua_ref determining others nonkey attributes does not hold, since flight was defined as a weak entity.

The relation flight is in BCNF since the unique existent determinant, composed of rec_num, rank, year_ref, and qua_ref is the primary key.

15 - Ext_num

Ext_num is dependent on aviator, which key attribute is rec_num.

```
rec_num, num_ext_num --> dt_incl_ext_num,
                         dt_excl_ext_num

rec_num, dt_incl_ext_num --> num_ext_num,
                             dt_excl_ext_num
```

The primary key for the relation ext_num is composed of rec_num and num_ext_num, because they can uniquely identify the relation, rec_num and dt_incl_ext_num are candidate keys for the relation, num_ext_num was selected to be the primary key, because has a value shorter than dt_incl_ext_num.

The relation ext_num is in 3NF, since each attribute has an atomic value, every nonkey attribute is fully dependent on rec_num and num_ext_num, and also every nonkey attribute is nontransitivelly dependent on rec_num and num_ext_num.

The relation ext_num is in BCNF since the existent determinants, (rec_num, num_ext_num) and (rec_num,dt_incl_ext_num) are candidate keys.

16 - Non_duty

Non_duty is dependent on unit, which key attribute is abbrev.

abbrev, dt_non_duty_sta --> dt_return, sum_non_duty_sta

The primary key for the relation non_duty is composed of abbrev and dt_non_duty_sta because they can uniquely identify the relation, no other attribute can be used as a candidate key for the relation.

The relation non_duty is in 3NF, since each attribute has an atomic value, every nonkey attribute is fully dependent on abbrev and dt_non_duty_sta, and also every nonkey attribute is nontransitively dependent on abbrev and dt_non_duty_sta.

The possible dependence dt_non_duty_sta determining others nonkey attributes does not hold, since non_duty was defined as a weak entity.

The relation non_duty is in BCNF since the unique existent determinant, composed of abbrev and dt_non_duty_sta is the primary key.

17 - Moving

Moving is dependent on unit and personnel, which key attributes are abbrev and recnum.

abbrev, recnum, dt_moving --> dt_pres, dt_detach, unit_
mov, sit_mov, sum_mov

The primary key for the relation moving is composed of abbrev, recnum, and dt_moving because they can uniquely identify the relation. The attributes dt_pres and dt_detach could not be used as candidate key, because during the moving the only date filled in is dt_moving, dt_pres and dt_detach will be filled in later on. No other attribute can be used as a candidate key for the relation.

The relation moving is in 3NF, since each attribute has an atomic value, every nonkey attribute is fully dependent on abbrev, recnum, and dt_moving, and also every nonkey attribute is nontransitively dependent on abbrev, recnum, and dt_moving.

The possible dependence dt_moving determining others nonkey attributes does not hold, since moving was defined as a weak entity

The relation moving is in BCNF since the unique existent determinant, composed of abbrev, recnum, and dt_moving is the primary key.

18 - Nomination

Nomination is dependent on unit and personnel, which key attributes are abbrev and recnum.

```
abbrev, recnum, dt_nom --> dt_exo, sit_unit_nom,
                           sum_nom
```

The primary key for the relation nomination is composed of abbrev, recnum and dt_nom because they can uniquely identify the relation. The attribute dt_exo could not be used as a candidate key, because it contains spaces during the nomination, only filled in at the end of the nomination. No other attribute can be used as a candidate key for the relation.

The relation nomination is in 3NF, since each attribute has an atomic value, every nonkey attribute is fully dependent on abbrev, recnum and dt_nom, and also every nonkey attribute is nontransitively dependent on abbrev, recnum and dt_nom.

The possible dependence dt_nom determining others nonkey attributes does not hold, since nomination was defined as a weak entity.

The relation nomination is in BCNF since the unique existent determinant, composed of abbrev, recnum and dt_nom is the primary key.

19 - Designation

Designation is dependent on unit and personnel, which key attributes are abbrev and recnum.

```
abbrev, recnum, dt_desig --> dt_waiver, sit_unit_des,
                             sum_des
```

The primary key for the relation designation is composed of abbrev, recnum and dt_desig because they can uniquely identify the relation. The attribute dt_waiver could not be considered candidate key, because is just filled in at the end of the designation. No other attribute can be used as a candidate key for the relation.

The relation designation is in 3NF, since each attribute has an atomic value, every nonkey attribute is fully dependent on abbrev, recnum and dt_desig, and also every nonkey attribute is nontransitively dependent on abbrev, recnum and dt_desig.

The possible dependence dt_desig determining others nonkey attributes does not hold, since designation was defined as a weak entity.

The relation designation is in BCNF since the unique existent determinant, composed of abbrev, recnum and dt_desig is the primary key.

20 - Attachment

Attachment is dependent on unit and personnel, which key attributes are abbrev and recnum.

```
abbrev, recnum, dt_start_att --> dt_end_att, sit_unit_
                                 att, sum_att, rea_att
```

The primary key for the relation attachment is composed of abbrev, recnum and dt_start_att because they can uniquely identify the relation. The attribute dt_end_att could not be used as candidate key, because is just filled in at the end of the attachment. No other attribute can be used as a candidate key for the relation.

The relation attachment is in 3NF, since each attribute has an atomic value, every nonkey attribute is fully dependent on abbrev, recnum and dt_start_att, and also every nonkey attribute is nontransitively dependent on abbrev, recnum and dt_start_att.

The possible dependence dt_start_att determining others nonkey attributes does not hold, since attachment was defined as a weak entity.

The relation attachment is in BCNF since the unique existent determinant, composed of abbrev, recnum and dt_start_att is the primary key.

Relationship --> Relation.

As explained in chapter IV, the process to transform a relationship between two or more entities, into a relation, is to get the key attribute or discriminator of each entity, to form the primary key of the relation, the nonkey attributes of the relationship will become nonkey attributes of the relation.

21 - Personnel_leave

Per_lea is the relationship between the entities personnel, which key attribute is rec_num, and leave, which key attribute is type_leave.

```
rec_num, type_leave --> dt_start_leave, dt_end_leave
rec_num, dt_start_leave --> type_leave, dt_end_leave
```

The primary key for the relation per_lea, is composed of rec_num and type_leave, because they can uniquely identify the relation, rec_num and dt_start_leave are candidate keys for the relation, type_leave was selected to be the primary key, because has a value shorter than dt_start_leave. No other attribute can be used as a candidate key for the relation.

The relation per_lea is in 3NF, since each attribute has an atomic value, every nonkey attribute is fully dependent on rec_num and type_leave, and also every nonkey attribute is nontransitively dependent on rec_num and type_leave.

The relation per_lea is in BCNF since the existent determinants, (rec_num, type_leave) and (rec_num,dt_start_leave) are candidate keys.

22 - Personnel_promotion

Per_pro is the relationship between the entities personnel, which key attribute is rec_num, and promotion which key attribute is rank.

```
rec_num, rank --> crit_pro, dt_pro
rec_num, dt_pro --> crit_pro, rank
```

The primary key for the relation per_pro, is composed of rec_num and rank, because they can uniquely identify the relation, rec_num and dt_pro are candidate keys for the relation, rank was selected to be the primary key, because has a value shorter than dt_pro. No other attribute can be used as a candidate key for the relation.

The relation per_pro is in 3NF, since each attribute has an atomic value, every nonkey attribute is fully dependent on rec_num and rank, and also every nonkey attribute is nontransitively dependent on rec_num and rank.

The relation per_pro is in BCNF since the existent determinants, (rec_num, rank) and (rec_num, dt_pro) are candidate keys.

23 - Personnel_course

Per_cou is the relationship between the entities personnel, which key attribute is rec_num, and course, which key attribute is type_course.

```
rec_num, type_course --> dt_start_course, dt_end_course,
                         grade_course, classif_course,
                         fin_course

rec_num, dt_start_course --> type_course, dt_end_course,
                             grade_course, classif_course,
                             fin_course
```

The primary key for the relation per_cou, is composed of rec_num and type_course, because they can uniquely identify the relation, rec_num and dt_start_course are candidate keys for the relation, type_course was selected to be the primary key, because has a value that can be more easily identified than dt_start_course. The attribute dt_end_course could not be used as a candidate key, for the reasons already explained in the relation course. No other attribute can be used as a candidate key for the relation.

The relation per_cou is in 3NF, since each attribute has an atomic value, every nonkey attribute is fully dependent on rec_num and type_course, and also every nonkey attribute is nontransitively dependent on rec_num and type_course.

The relation per_cou is in BCNF since the existent determinants, (rec_num, type_course) and (rec_num,dt_start_ course) are candidate keys.

24 - Medal

Per_med is the relationship between the entities personnel, which key attribute is rec_num, and medal, which key attribute is type_medal.

rec_num, type_medal --> dt_medal

The primary key for the relation per_med, is composed of rec_num and type_medal, because they can uniquely identify the relation, no other attribute can be used as a candidate key for the relation.

The relation per_med is in 3NF, since each attribute has an atomic value, every nonkey attribute is fully dependent on rec_num and type_medal, and also every nonkey attribute is nontransitively dependent on rec_num and type_medal.

The relation per_med is in BCNF since the unique existent determinant, composed of rec_num and type_medal is the primary key.

25 - Personnel_exclusion

Per_exc is the relationship between the entities personnel, which key attribute is rec_num, and exclusion, which key attribute is rea_excl.

```
rec_num, rea_excl --> dt_excl
rec_num, dt_excl --> rea_excl
```

The primary key for the relation per_exc, is composed of rec_num and rea_excl, because they can uniquely identify the relation, rec_num and dt_excl are candidate keys for the relation, rea_excl was selected to be the primary key, because has a value shorter than dt_excl. No other attribute can be used as a candidate key for the relation.

The relation per_exc is in 3NF, since each attribute has an atomic value, every nonkey attribute is fully dependent on rec_num and rea_excl, and also every nonkey attribute is nontransitively dependent on rec_num and rea_excl.

The relation per_exc is in BCNF since the existent determinants, (rec_num, rea_excl) and (rec_num, dt_excl) are candidate keys.

26 - Personnel_death

Per_dea is the relationship between the entities personnel, which key attribute is rec_num, and death, which key attribute is rea_excl.

```
rec_num, rea_excl --> dt_excl, cause_death
rec_num, dt_excl --> rea_excl, cause_death
rec_num, cause_death --> dt_excl, rea_excl
```

The primary key for the relation per_dea, is composed of rec_num and rea_excl, because they can uniquely identify the relation, (rec_num, dt_excl) and (rec_num, cause_death) are candidate keys for the relation, rea_excl was selected to be the primary key, because has a value that can be more easily identified than dt_excl and cause_death. No other attribute can be used as a candidate key for the relation.

The relation per_dea is in 3NF, since each attribute has an atomic value, every nonkey attribute is fully dependent on rec_num and rea_excl, and also every nonkey attribute is nontransitively dependent on rec_num and rea_excl.

The relation per_dea is in BCNF since the existent determinants, (rec_num, rea_excl), (rec_num, dt_excl) and (rec_num, cause_death) are candidate keys.

27 - Personnel_inclusion

Per_inc is the relationship between the entities personnel, which key attribute is rec_num, and inclusion, which key attribute is rea_incl.

```
rec_num, rea_incl --> dt_incl
rec_num, dt_incl --> rea_incl
```

The primary key for the relation per_inc, is composed of rec_num and rea_incl, because they can uniquely identify the relation, rec_num and dt_incl are candidate keys for the relation, rea_incl was selected to be the primary key, because has a value shorter than dt_incl. No other attribute can be used as a candidate key for the relation.

The relation per_inc is in 3NF, since each attribute has an atomic value, every nonkey attribute is fully dependent on rec_num and rea_incl, and also every nonkey attribute is nontransitively dependent on rec_num and rea_incl.

The relation per_inc is in BCNF since the existent determinants, (rec_num, rea_incl) and (rec_num, dt_incl) are candidate keys.

28 - Personnel_specialty

Per_spe the relationship between the entities personnel, which key attribute is rec_num, and specialty, which key attribute is active_list.

```
rec_num, active_list --> dt_incl_act_list
rec_num, dt_incl_act_list --> active_list
```

The primary key for the relation per_spe, is composed of rec_num and active_list, because they can uniquely identify the relation, rec_num and dt_incl_act_list are candidate keys for the relation, active_list was selected to be the primary key, because has a value shorter than dt_incl_act_list. No other attribute can be used as a candidate key for the relation.

The relation per_spe is in 3NF, since each attribute has an atomic value, every nonkey attribute is fully dependent on rec_num and active_list, and also every nonkey attribute is nontransitively dependent on rec_num and active_list.

The relation per_spe is in BCNF since the existent determinants, (rec_num, active_list) and (rec_num, dt_incl_act_list) are candidate keys.

29 - Personnel_unit.

Per_unit is the relationship between entities personnel, which key attribute is rec_num, and unit, which key attribute is abbrev.

The primary key for the relation is composed of rec_num, and abbrev. Since there is no attribute other than the primary key, thc relation is clearly in 3NF and BCNF.

30 - Aviator_promotion_flight

Avi_pro_fli is the relationship between entities aviator, which key attribute is rec_num, promotion, which key attribute is rank, and flight, which discriminators are year_ref and qua_ref.

The primary key for the relation is composed of rec_num, rank, year_ref, and qua_ref. Since there is no attribute other than the primary key, the relation is clearly in 3NF and BCNF.

31 - Aviator_ifr_card

Avi_ifr is the relationship between entities aviator, which key attribute is rec_num, and ifr_card, which key attribute is num_ifr_card.

The primary key for the relation is composed of rec_num, and num_ifr_card. Since there is no attribute other than the primary key, the relation is clearly in 3NF and BCNF.

32 - Aviator_ext_num

Avi_ext is the relationship between entities aviator, which key attribute is rec_num, and ext_num, which discriminator is num_ext_num.

The primary key for the relation is composed of rec_num, and num_ext_num. Since there is no attribute other than the primary key, the relation is clearly in 3NF and BCNF.

33 - Personnel_stability.

Per_sta is the relationship between entities personnel, which key attribute is rec_num, and stability, which discriminator is num_stabil.

The primary key for the relation is composed of rec_num, and num_stabil. Since there is no attribute other than the primary key, the relation is clearly in 3NF and BCNF.

34 - Personnel_unit_moving

Per_unit_mov is the relationship between entities personnel, which key attribute is rec_num, unit, which key attribute is abbrev, and moving, which discriminator is dt_moving.

The primary key for the relation is composed of rec_num, abbrev, and dt_moving. Since there is no attribute other than the primary key, the relation is clearly in 3NF and BCNF.

35 - Personnel_unit_nomination

Per_unit_nom is the relationship between entities personnel, which key attribute is rec_num, unit, which key attribute is abbrev, and nomination, which discriminator is dt_nom.

The primary key for the relation is composed of rec_num, abbrev, and dt_nom. Since there is no attribute other than the primary key, the relation is clearly in 3NF and BCNF.

36 - Personnel_unit_designation

Per_unit_des is the relationship between entities personnel, which key attribute is rec_num, unit, which key attribute is rank, and designation, which discriminator is dt_desig.

The primary key for the relation is composed of rec_num, abbrev, and dt_desig. Since there is no attribute other than the primary key, the relation is clearly in 3NF and BCNF.

37 - Personnel_unit_attachment

Per_unit_att is the relationship between entities personnel, which key attribute is rec_num, unit, which key attribute is abbrev, and attachment, which discriminator is dt_start_att.

The primary key for the relation is composed of rec_num, abbrev, and dt_start_att. Since there is no attribute other than the primary key, the relation is clearly in 3NF and BCNF.

38 - Personnel_specialty_non_duty

Per_spe_non is the relationship between entities personnel, which key attribute is rec_num, specialty, which key attribute is active_list, and non_duty, which discriminator is dt_non_duty_sta.

The primary key for the relation is composed of rec_num, active_list, and dt_non_duty_sta. Since there is no attribute other than the primary key, the relation is clearly in 3NF and BCNF.

39 - Unit_promotion_specialty

Unit_pro_spe is the relationship between entities unit, which key attribute is abbrev, promotion, which key attribute is rank, and specialty, which key attribute is active_list.

```
abbrev, rank, active_list --> pred_rank_spe,
                              eff_rank_spe
```

The primary key for the relation unit_pro_spe is composed of abbrev, rank, and active_list, because they can uniquely identify the relation, no other attribute can be used as a candidate key for the relation.

The relation unit_pro_spe is in 3NF, since each attribute has an atomic value, every nonkey attribute is fully dependent on abbrev, rank, and active_list, and also every nonkey attribute is nontransitively dependent on abbrev, rank, and active_list.

The relation unit_pro_spe is in BCNF since the unique existent determinant, composed of abbrev, rank, and active_list is the primary key.

Appendix E

Prototype Programs

This Appendix shows two examples of programs written for the prototype of the Database for the Brazilian Air Force, Mainmenu and Subsel. Figure E-1 shows the structure of the designed prototype programs.

```
                        +----------+
                        |          |
                        | MAINMENU |
                        |          |
                        +----------+
                             |
                             |
       +-------------+-------------+-------------+
       |             |             |             |
       |             |             |             |
  +---------+    +--------+    +--------+    +--------+
  |         |    |        |    |        |    |        |
  | SUBINCL |    | SUBSEL |    | SUBMOD |    | SUBDEL |
  |         |    |        |    |        |    |        |
  +---------+    +--------+    +--------+    +--------+
```

Figure E-1 - Prototype Programs' Structure

Mainmenu is the program that controls the entire prototype, by selecting an option, the user tell mainmenu which subprogram to call.

Subincl is a program to handle inclusions and additions to the database. Subsel is called to allows the user to make some predefined seletions in the database. Submod is called to handle modifications in the database. Subdel handles deletions of wrong information included in the database.

```
 IDENTIFICATION DIVISION.
 PROGRAM-ID. MAINMENU.
 AUTHOR.
     WAGNER MUSSATO, MAJ AV
     BRAZILIAN AIR FORCE.
*
*    DESCRIPTION OF THE PROGRAM
*
*    MAINMENU IS THE MAIN PROGRAM OF A SET OF PROGRAMS DEVELOPED,
*                TO BE USED AS A PROTOTYPE OF THE PERSONNEL DATA-
*                BASE SYSTEM FOR THE BRAZILIAN AIR FORCE.
*
*    LANGUAGE  : COBOL
*
*    DBMS      : ORACLE
*
*    ALGORITHM : SHOW SCREEN
*                ASK FOR ONE SELECTION
*                CHECK THE SELECTION MADE
*                IF CORRECT
*                    LOGON ORACLE
*                    CALL SUBPROGRAMS ACCORDING TO THE SELECTION
*                IF NOT CORRECT
*                    ASK FOR ANOTHER SELECTION
*                LOGOFF ORACLE
*                END OF PROGRAM
*
 ENVIRONMENT DIVISION.
 CONFIGURATION SECTION.
 SOURCE-COMPUTER. HARRIS.
 OBJECT-COMPUTER. HARRIS.
 DATA DIVISION.
 WORKING-STORAGE SECTION.
*
*    VARIABLES USED FOR THE ORACLE DATABASE
*
 77  ERR-RC                PIC S9999 COMP.
 77  ERR-RCX               PIC S9999 SIGN LEADING SEPARATE DISPLAY.
 77  ERR-FUNC              PIC S9999 SIGN LEADING SEPARATE DISPLAY.
 77  STATUS-DISPLAY        PIC S9(7) SIGN LEADING SEPARATE DISPLAY.
 77  MSGBUF                PIC X(80).
 77  DCM-STAT              PIC S9999 VALUE 0 COMP-1.
 77  C-FNC                 PIC S9999 COMP.
 77  C-RC                  PIC S9999 COMP.
 77  EIGHTY                PIC S9999 VALUE 80 COMP.
 77  CURSOR-SIZE           PIC S9999 VALUE 5000 COMP.
 77  STAT                  PIC S9999 VALUE 0    COMP.
 77  DBASE-NAME            PIC X(6)  VALUE "ORACLE".
 77  DBASE-NAME-LENGTH     PIC S9999 VALUE 6 COMP.
 77  USER-ID               PIC X(7) VALUE "MUSSATO".
 77  USER-ID-LENGTH        PIC S9999 VALUE 7 COMP.
 77  PASSWORD              PIC X(7)  VALUE "BRAZIL".
 77  PASSWORD-LENGTH       PIC S9999 VAI UE 6 COMP.
 77 AUDIT-OFF           PIC S9999 VALUE 0 COMP.
 77 DCM-VALUE           PIC S9999 VALUE 70 COMP-1.
*
*   SCREEN DEFINITION
*
 77 BAF                 PIC X(50) VALUE
    "               BRAZILIAN  AIR  FORCE".
 77 PDL                 PIC X(50) VALUE
    "                 PERSONNEL DATABASE".
 77 MAIN                PIC X(50) VALUE
    "                    MAIN MENU".
 77 INADD               PIC X(50) VALUE
    "               1.  INCLUDE / ADD".
 77 SEL                 PIC X(50) VALUE
    "               2.  SELECT".
 77 DEL                 PIC X(50) VALUE
    "               3.  DELETE".
 77 MOD                 PIC X(50) VALUE
    "               4.  MODIFY".
 77 QUIT                PIC X(50) VALUE
    "               5.  QUIT".
 77 HELP                PIC X(50) VALUE
    "               6.  HELP".
 77 SELONE              PIC X(50) VALUE
    "       SELECT ONE:".
*
*   RESP = USED TO RECEIVE THE USER ANSWER
 77 RESP                PIC XX JUST RIGHT.
*
*   FLORA = FLAG TO CONTROL ORACLE LOGON
 77 FLORA               PIC 9 VALUE ZEROS.
*
 PROCEDURE DIVISION.
 BEGIN.
*
*   SHOW SCREEN
*
    DISPLAY BAF.
    DISPLAY " ".
    DISPLAY PDB.
    DISPLAY " ".
    DISPLAY MAIN.
    DISPLAY " ".    DISPLAY " ".
    DISPLAY INADD.
    DISPLAY " ".
    DISPLAY SEL.
    DISPLAY " ".
    DISPLAY DEL.
    DISPLAY " ".
    DISPLAY MOD.
    DISPLAY " ".
    DISPLAY QUIT.
    DISPLAY " ".
    DISPLAY HELP.
    DISPLAY " ".
    DISPLAY SELONE.
*
*   RECEIVE THE ANSWER
    ACCEPT RESP.
*
*   CHECK IF ANSWER IS CORRECT
*
    IF RESP NOT = "1" AND "2" AND "3" AND "4" AND "5" AND "6"
       DISPLAY RESP " WRONG ANSWER, TRY AGAIN"
       ACCEPT RESP
       IF RESP NOT = "1" AND "2" AND "3" AND "4" AND "5" AND "6"
           DISPLAY RESP " STILL WRONG END OF PGM"
           STOP RUN.
*
*   CHECK IF ORACLE IS ALREADY LOGGED ON
*
    IF FLORA = 0
        DISPLAY "LOGGING ON ORACLE - PLEASE WAIT.."
        PERFORM LOG-ORA THRU END-LOG-ORA
        MOVE 1 TO FLORA.
*
*   SELECT THE PROGRAM TO CALL, ACCORDING TO "RESP"
*
    IF RESP = "1" CALL "SUBINCL"  ELSE
       IF RESP = "2" CALL "SUBSEL" ELSE
          IF RESP = "3" CALL "SUBDEL" ELSE
             IF RESP = "4" CALL "SUBMOD"  ELSE
                IF RESP = "5" PERFORM QUITMM ELSE
                    IF RESP = "6" PERFORM HELPMM.
    GO TO BEGIN.
*
*
 LOG-ORA.
* ALLOCATE DCM
*
      CALL "CPIDCM" USING DCM-VALUE, DCM-STAT.
*
* LOGON TO ORACLE
*
      CALL "CPIINI" USING DBASE-NAME, DBASE-NAME-LENGTH, STAT.
*
      CALL "CPILON" USING USER-ID, USER-ID-LENGTH, PASSWORD,
            PASSWORD-LENGTH, AUDIT-OFF, STAT.
      IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-STOP.
* OPEN CURSOR
*
        CALL "CPIOPN" USING C-RC, CURSOR-SIZE, STAT.
        IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-LOGOF.
*
* DISABLE AUTO-COMMIT
*
        CALL "CPICOF" USING STAT.
         IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
```

```
            DISPLAY "LOGGED TO ORACLE".
 END-LOG-ORA.
     EXIT.
*
*    END OF PROGRAM
*
 QUITMM.
     DISPLAY "QUIT FROM MAIN MENU".
     STOP RUN.
 HELPMM.
     DISPLAY "HELP MAIN MENU".
*
*
 EXIT-CLOSE.
     CALL "CPICLS" USING C-RC, STAT.
     IF STAT NOT = 0 PERFORM ORA-ERR.
*
*
 EXIT-LOGOF.
     CALL "CPILOF" USING STAT.
     IF STAT NOT = 0 PERFORM ORA-ERR.
*
 END-OF-PGM.
 EXIT-STOP.
     EXIT PROGRAM. STOP-RUN.
*
*    DISPLAY ORACLE ERROR NOTICE
*
 ORA-ERR.
     IF STAT NOT = 0 MOVE STAT TO ERR-RC
              MOVE "0" TO ERR-FUNC
     ELSE IF C-RC NOT = 0 MOVE C-RC TO ERR-RC MOVE C-FNC TO
          ERR-FUNC
          ELSE MOVE C-RC TO ERR-RC MOVE C-FNC TO ERR-FUNC.
     MOVE ERR-RC TO ERR-RCX.
     DISPLAY "ORACLE ERROR. CODE IS ", ERR-RCX, ",FUNCTION IS ",
          ERR-FUNC.
     CALL "CPIGEM" USING ERR-RC, MSGBUF, EIGHTY, STAT.
     DISPLAY MSGBUF.
*



 ********     S U B S E L     **********


 IDENTIFICATION DIVISION.
 PROGRAM-ID. SUBSEL.
 AUTHOR.
     WAGNER MUSSATO, MAJ AV
     BRAZILIAN AIR FORCE.
*
*    DESCRIPTION OF THE PROGRAM
```

```
*
*    SUBSEL   IS THE SUBROUTINE CALLED FROM MAINMENU, THAT ALLOWS
*                THE USER TO SELECT MOST OF THE DATA EXISTING ON
*                THE PERSONNEL DATABASE.
*
*    LANGUAGE  : COBOL
*
*    DBMS      : ORACLE
*
*    ALGORITHM : SHOW SCREEN
*                ASK FOR ONE SELECTION
*                CHECK THE SELECTION MADE
*                IF CORRECT
*                    LOGON ORACLE
*                    CALL SUBROUTINES ACCORDING TO THE SELECTION
*                IF NOT CORRECT
*                    ASK FOR ANOTHER SELECTION
*                RETURN TO MAIN PROGRAM
*
 ENVIRONMENT DIVISION.
 CONFIGURATION SECTION.
 SOURCE-COMPUTER. HARRIS.
 OBJECT-COMPUTER. HARRIS.
 DATA DIVISION.
 WORKING-STORAGE SECTION.
*
*    SCREEN DEFINITION
*
 77  BAF               PIC X(50) VALUE
     "                 BRAZILIAN  AIR  FORCE".
 77  PDB               PIC X(50) VALUE
     "                   PERSONNEL DATABASE".
 77  SELAV             PIC X(70) VALUE
     "                  SELECTIONS AVAILABLE".
 77  SEL1              PIC X(70) VALUE
     "       1.  GIVEN A TAG NAME GET RECNUM, ACTLIST, RANK, UNIT"
     .
 77  SEL2              PIC X(70) VALUE
     "       2.  GIVEN A RECNUM GET PERSONNAL INFORMATIONS".
 77  SEL3              PIC X(70) VALUE
     "       3.  GIVEN A RECNUM GET MOVING HISTORICAL".
 77  SEL4              PIC X(70) VALUE
     "       4.  GIVEN A RECNUM GET NOMINATION HISTORICAL".
 77  SEL5              PIC X(70) VALUE
     "       5.  GIVEN A RECNUM GET DESIGNATION  HISTORICAL".
 77  SEL6              PIC X(70) VALUE
     "       6.  GIVEN A RECNUM GET ATTACHMENT HISTORICAL".
 77  SEL7              PIC X(70) VALUE
     "       7.  GIVEN A RECNUM GET FLIGHT INFORMATIONS".
 77  SEL8              PIC X(70) VALUE
     "       8.  GIVEN AN UNIT GET RELATED INFORMATIONS".
 77  SEL9              PIC X(70) VALUE
     "       9.  GIVEN AN UNIT GET PERSONS ASSIGNED".
 77  SEL10             PIC X(70) VALUE
```

```
      "       10.  SELECT UNITS WITH EXCEDENTS (EFFECT > PRED)".
 77   CUIT              PIC X(30) VALUE
      "11.    QUIT".
 77   RELP              PIC X(30) VALUE
      "12.    HELP".
 77   SELUM             PIC X(30) VALUE
      "SELECT ONE:".
*
*     VARIABLES USED IN THE PROGRAM STORED IN LIBRARY
*
 COPY PGMVAR OF LIBTES.
*
*     LIST OF THE SELECT ORACLE COMMANDS USED IN THE PROGRAM
*
 77   SQL-SEL1          PIC X(150) VALUE
      "SELECT RECNUM,CACTLIST,CRANK,CUNIT FROM PERSONNEL WHERE
-     "TNAME = :TNAME".
 77   SQL-SEL1-LENGTH    PIC S9999 VALUE 150 COMP.
*
 77   SQL-SEL2          PIC X(150) VALUE
      "SELECT STBIRTH,TNAME,UNITH,DTEXPH,CRANK,CACTLIST,CUNIT FROM
-     "PERSONNEL WHERE RECNUM = :RECNUM".
 77   SQL-SEL2-LENGTH    PIC S9999 VALUE 150 COMP.
*
 77   SQL-SEL3          PIC X(150) VALUE
      "SELECT ABBREV,DTMOV,DTPRES,DTDETACH,SITU FROM MOVING WHERE
-     "RECNUM = :RECNUM".
 77   SQL-SEL3-LENGTH    PIC S9999 VALUE 150 COMP.
*
 77   SQL-SEL4         PIC X(150) VALUE
      "SELECT ABBREV,DTNOM,DTEXO,SITN FROM NOMINATION WHERE RECNUM
-     "= :RECNUM".
 77   SQL-SEL4-LENGTH    PIC S9999 VALUE 150 COMP.
*
 77   SQL-SEL5          PIC X(150) VALUE
      "SELECT ABBREV,DTDES,DTWAIVER,SITD FROM DESIG WHERE RECNUM
-     "= :RECNUM".
 77   SQL-SEL5-LENGTH    PIC S9999 VALUE 70 COMP.
*
 77   SQL-SEL6            PIC X(150) VALUE
      "SELECT ABBREV,DTSATT,DTEATT,SITA FROM ATTACH WHERE RECNUM
-     "= :RECNUM".
 77   SQL-SEL6-LENGTH    PIC S9999 VALUE 150 COMP.
*
 77   SQL-SEL7           PIC X(150) VALUE
      "SELECT RANK,YEAREF,QUAREF,P1DIU,P2DIU,P1NOC,P2NOC FROM
-     "FLIGHT WHERE RECNUM =:RECNUM".
 77   SQL-SEL7-LENGTH    PIC S9999 VALUE 150 COMP.
*
 77   SQL-SEL8           PIC X(150) VALUE
      "SELECT UNAME,LOCAL,REGCOM,MAJCOM FROM UNIT WHERE ABBREV =
-     ":ABBREV".
 77   SQL-SEL8-LENGTH    PIC S9999 VALUE 150 COMP.
*
```

```
 77  SQL-SEL9            PIC X(150) VALUE
     "SELECT RECNUM,TNAME FROM PERSONNEL WHERE CUNIT =:ABBREV".
 77  SQL-SEL9-LENGTH     PIC S9999 VALUE 150 COMP.
*
 77  SQL-SEL10           PIC X(150) VALUE
     "SELECT ABBREV,RANK,ACTLIST,EFFECT,PRED FROM UNITPROSPE
-    " WHERE EFFECT > PRED".
 77  SQL-SEL10-LENGTH    PIC S9999 VALUE 150 COMP.
*
*    DISP-ONE = CONSTANT DISPLAYED IN THE PROGRAM
 77  DISP-ONE            PIC X(70) VALUE
     "     RECNUM      ACTLIST       RANK      UNIT".
*
 PROCEDURE DIVISION.
*
*    OPEN ORACLE CURSOR, DEFINE AREA TO BE USED BY ORACLE
*
 BEGIN.
     PERFORM OPENCUR THRU ENDCUR.
*
*    SHOW THE SCREEN
*
 SELECTION.
     DISPLAY BAF.    DISPLAY " ".
     DISPLAY PDB.    DISPLAY " ".
     DISPLAY SELAV.  DISPLAY " ".
     DISPLAY SEL1.
     DISPLAY SEL2.
     DISPLAY SEL3.
     DISPLAY SEL4.
     DISPLAY SEL5.
     DISPLAY SEL6.
     DISPLAY SEL7.
     DISPLAY SEL8.
     DISPLAY SEL9.
     DISPLAY SEL10.
     DISPLAY CUIT.
     DISPLAY RELP.
     DISPLAY SELUM.
*
*    RECEIVE THE USER SELECTION
*
     ACCEPT RESP.
*
*    CHECK IF SELECTION IS CORRECT
*
     IF NOT RESP-OK
         DISPLAY "RESP --> ", RESP, " WRONG RESP TRY AGAIN"
         ACCEPT RESP
         IF NOT RESP-OK
            DISPLAY "RESP --> ", RESP, " STILL WRONG END OF PMG"
            EXIT PROGRAM.
*
*    SELECT THE ROUTINE ACCORDING TO THE USER SELECTION
```

```
*
     IF RESP = "1" PERFORM SELONE THRU END-ONE ELSE
      IF RESP = "2" PERFORM SELTWO THRU END-TWO ELSE
       IF RESP = "3" PERFORM SELTHREE THRU END-THREE ELSE
        IF RESP = "4" PERFORM SELFOUR THRU END-FOUR ELSE
         IF RESP = "5" PERFORM SELFIVE THRU END-FIVE ELSE
          IF RESP = "6" PERFORM SELSIX THRU END-SIX  ELSE
           IF RESP = "7" PERFORM SELSEVEN THRU END-SEVEN ELSE
            IF RESP = "8" PERFORM SELEIGHT THRU END-EIGHT ELSE
             IF RESP = "9" PERFORM SELNINE THRU END-NINE ELSE
              IF RESP = "10" PERFORM SELTEN THRU END-TEN ELSE
               IF RESP = "12" PERFORM HELP  THRU END-HELP ELSE
                IF RESP = "11" PERFORM QUIT  THRU END-QUIT ELSE
                DISPLAY "NOT IMPLEMENTED".
     GO TO SELECTION.
*
* OPEN CURSOR
*
 OPENCUR.
         CALL "CPIOPN" USING C-RC, CURSOR-SIZE, STAT.
         IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-LOGOF.
 ENDCUR.  EXIT.
*
*    SELECT ONE = GIVEN A TAG NAME GET RECNUM, ACTLIST, RANK, UNIT
*
 SELONE.
     DISPLAY "ENTER TAG NAME (20):".
     ACCEPT TNAME.
     IF TNAME = " "
         DISPLAY "TAG NAME CAN NOT BE NULL"
         DISPLAY "DO YOU WANT TO ABORT THIS SELECTION (Y/N)?"
         ACCEPT Y-N
         IF Y-N = "Y" OR "y"
             GO TO END-ONE
         ELSE
             GO TO SELONE.
*
* ***    NAMES USED TO ISSUED A CALL TO ORACLE      ***
*
*    CPIEXE    = EXECUTING A SQL COMMAND
*
*    CPIBVC    = BIND SQL SUBSTITUTION CHARACTER VARIABLES
*
*    CPIBVN    = BIND SQL SUBSTITUTION NUMERIC VARIABLES
*
*    CPIDFC    = DEFINE A CHARACTER RESULT BUFFER
*
*    CPIDFN    = DEFINE A NUMERIC RESULT BUFFER
*
*    CPIFCH    = RETRIEVING A ROW OF A RESULT TABLE
*
     CALL "CPIOSQ" USING C-RC, SQL-SEL1, SQL-SEL1-LENGTH,
     STAT.
     IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
```

```
BVC-TNAME.
    CALL "CPIBVC" USING C-RC, TNAME-N, TNAME-N-LENGTH,
        TNAME, TNAME-LENGTH, ASC, STAT.
    IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
    MOVE 2 TO POS.
DFC-CACTLIST.
    CALL "CPIDFC" USING C-RC, POS, CACTLIST, CACTLIST-LENGTH, ASC,
         SCALE, STAT.
    IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
    MOVE 1 TO POS.
DFC-RECNUM.
    CALL "CPIDFC" USING C-RC, POS, RECNUM, RECNUM-LENGTH, ASC,
         SCALE, STAT.
    IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
    MOVE 3 TO POS.
DFC-CRANK.
    CALL "CPIDFC" USING C-RC, POS, CRANK, CRANK-LENGTH,
        ASC, SCALE, STAT.
    IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
    MOVE 4 TO POS.
DFC-CUNIT.
    CALL "CPIDFC" USING C-RC, POS, CUNIT, CUNIT-LENGTH,
        ASC, SCALE, STAT.
    IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
CALLEXE.
    CALL "CPIEXE" USING C-RC, STAT.
    IF STAT NOT = 0 AND 1403 PERFORM ORA-ERR GO TO EXIT-CLOSE.
ENDEXE.
    DISPLAY DISP-ONE.
    DISPLAY " ".
ONEFCH.
    PERFORM CALLFCH.
    IF STAT = 1403
        GO TO END-ONE.
    MOVE RECNUM TO RECNUM-W.
    DISPLAY "    ", RECNUM-W, "     ", CACTLIST, "     ", CRANK,
            "     ", CUNIT.
    GO TO ONEFCH.
END-ONE.
    PERFORM PAUSA THRU END-PAUSA.
*
*    SELECT TWO = GIVEN A RECNUM GET PERSONNEL INFORMATIONS
*
SELTWO.
    DISPLAY "ENTER RECNUM (12):".
    ACCEPT RECNUM.
    IF RECNUM = " "
        DISPLAY "RECNUM CAN NOT BE NULL"
        DISPLAY "DO YOU WANT TO ABORT THIS SELECTION (Y/N)?"
        ACCEPT Y-N
        IF Y-N = "Y" OR "y"
            GO TO FIM-TWO
        ELSE
            GO TO SELTWO.
    CALL "CPIOSQ" USING C-RC, SQL-SEL2, SQL-SEL2-LENGTH,
    STAT.
    IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
 BVC-RECNUM.
    CALL "CPIBVC" USING C-RC, RECNUM-N, RECNUM-N-LENGTH,
        RECNUM, RECNUM-LENGTH, ASC, STAT.
    IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
    MOVE 1 TO POS.
 DFC-STBIRTH.
    CALL "CPIDFC" USING C-RC, POS, STBIRTH, STBIRTH-LENGTH,
        ASC, SCALE, STAT.
    IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
    MOVE 2 TO POS.
 DFC-TNAME.
    CALL "CPIDFC" USING C-RC, POS, TNAME, TNAME-LENGTH,
        ASC, SCALE, STAT.
    IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
    MOVE 3 TO POS.
 DFC-UNITH.
    CALL "CPIDFC" USING C-RC, POS, UNITH, UNITH-LENGTH,
        ASC, SCALE, STAT.
    IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
    MOVE 4 TO POS.
 DFC-DTEXPH.
    CALL "CPIDFC" USING C-RC, POS, DTEXPH, DTEXPH-LENGTH,
        ASC, SCALE, STAT.
    IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
    MOVE 5 TO POS. PERFORM DFC-CRANK.
    MOVE 6 TO POS. PERFORM DFC-CACTLIST.
    MOVE 7 TO POS. PERFORM DFC-CUNIT.
    PERFORM CALLEXE.
 CALLFCH.
    CALL "CPIFCH" USING C-RC, STAT.
    IF STAT NOT = 0 AND 1403 PERFORM ORA-ERR GO TO EXIT-CLOSE.
 END-FCH.
    IF STAT = 0
        DISPLAY " "
        DISPLAY "TNAME                STBIRTH   UNITH     DTEXPH
-        " RANK    ACTLIST      UNIT"
        DISPLAY " "
        DISPLAY  TNAME STBIRTH "   " UNITH "     " DTEXPH
        "     " CRANK "     " CACTLIST "    " CUNIT
    ELSE
        DISPLAY "RECNUM COULD NOT BE FOUND"
        DISPLAY "DO YOU WANT TO TRY AGAIN (Y/N)?"
        ACCEPT Y-N
        IF Y-N = "Y" OR "y"
            GO TO SELTWO.
    GO TO END-TWO.
 FIM-TWO.
    DISPLAY "SELECTION ABORTED".
 END-TWO.
    PERFORM PAUSA THRU END-PAUSA.
*
```

```
*    SELECT THREE = GIVEN A RECNUM GET MOVING HISTORICAL
*
 SELTHREE.
     DISPLAY "ENTER RECNUM (12):".
     ACCEPT RECNUM.
     IF RECNUM = " "
         DISPLAY "RECNUM CAN NOT BE NULL"
         DISPLAY "DO YOU WANT TO ABORT THIS SELECTION (Y/N)?"
         ACCEPT Y-N
         IF Y-N = "Y" OR "y"
             GO TO FIM-THREE
         ELSE
             GO TO SELTHREE.
     CALL "CPIOSQ" USING C-RC, SQL-SEL3, SQL-SEL3-LENGTH,
     STAT.
     IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
     PERFORM BVC-RECNUM.
     MOVE 1 TO POS.
 DFC-ABBREV.
     CALL "CPIDFC" USING C-RC, POS, ABBREV, ABBREV-LENGTH,
         ASC, SCALE, STAT.
     IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
     MOVE 2 TO POS.
 DFC-DTMOV.
     CALL "CPIDFC" USING C-RC, POS, DTMOV, DTMOV-LENGTH,
         ASC, SCALE, STAT.
     IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
     MOVE 3 TO POS.
 DFC-DTPRES.
     CALL "CPIDFC" USING C-RC, POS, DTPRES, DTPRES-LENGTH,
         ASC, SCALE, STAT.
     IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
     MOVE 4 TO POS.
 DFC-DTDETACH.
     CALL "CPIDFC" USING C-RC, POS, DTDETACH, DTDETACH-LENGTH,
         ASC, SCALE, STAT.
     IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
     MOVE 5 TO POS.
 DFC-SITU.
     CALL "CPIDFC" USING C-RC, POS, SITU, SITU-LENGTH,
         ASC, SCALE, STAT.
     IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
 END-SITU.
     PERFORM CALLEXE.
     DISPLAY " ".
     DISPLAY "ABBREV      DTMOV        DTPRES        DTDETACH
-    "  SITU".
     DISPLAY " ".
 ASKTHREE.
     PERFORM CALLFCH.
     IF STAT = 1403
         GO TO END-THREE
     ELSE
         DISPLAY ABBREV "    " DTMOV "    " DTPRES "    " DTDETACH
```

```
        "    " SITU
        GO TO ASKTHREE.
FIM-THREE.
    DISPLAY "SELECTION ABORTED".
END-THREE.
    PERFORM PAUSA THRU END-PAUSA.
*
*   SELECT FIVE = GIVEN A RECNUM GET DESIGNATION HISTIORICAL
*
SELFIVE.
    DISPLAY "ENTER RECNUM (12):".
    ACCEPT RECNUM.
    IF RECNUM = " "
        DISPLAY "RECNUM CAN NOT BE NULL"
        DISPLAY "DO YOU WANT TO ABORT THIS SELECTION (Y/N)?"
        ACCEPT Y-N
        IF Y-N = "Y" OR "y"
            GO TO FIM-FIVE
        ELSE
            GO TO SELFIVE.
    CALL "CPIOSQ" USING C-RC, SQL-SEL5, SQL-SEL5-LENGTH,
    STAT.
    IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
    PERFORM BVC-RECNUM.
    MOVE 1 TO POS. PERFORM DFC-ABBREV.
    MOVE 2 TO POS.
DFC-DTDES.
    CALL "CPIDFC" USING C-RC, POS, DTDES, DTDES-LENGTH,
        ASC, SCALE, STAT.
    IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
    MOVE 3 TO POS.
DFC-DTWAIVER.
    CALL "CPIDFC" USING C-RC, POS, DTWAIVER, DTWAIVER-LENGTH,
        ASC, SCALE, STAT.
    IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
    MOVE 4 TO POS.
DFC-SITD.
    CALL "CPIDFC" USING C-RC, POS, SITD, SITD-LENGTH,
        ASC, SCALE, STAT.
    IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
END-SITD.
    PERFORM CALLEXE.
    DISPLAY " ".
    DISPLAY "ABBREV      DTDES         DTDETACH        SITD".
    DISPLAY " ".
ASKFIVE.
    PERFORM CALLFCH.
    IF STAT = 1403
        GO TO END-FIVE
    ELSE
        DISPLAY ABBREV "    " DTDES "    " DTWAIVER "    " SITD
        GO TO ASKFIVE.
FIM-FIVE.
    DISPLAY "SELECTION ABORTED".
```

```
       END-FIVE.
           PERFORM PAUSA THRU END-PAUSA.
      *
      *    SELECT FOUR = GIVEN A RECNUM GET NOMINATION HISTORICAL
      *
       SELFOUR.
           DISPLAY "ENTER RECNUM (12):".
           ACCEPT RECNUM.
           IF RECNUM = " "
               DISPLAY "RECNUM CAN NOT BE NULL"
               DISPLAY "DO YOU WANT TO ABORT THIS SELECTION (Y/N)?"
               ACCEPT Y-N
               IF Y-N = "Y" OR "y"
                   GO TO FIM-FOUR
               ELSE
                   GO TO SELFOUR.
           CALL "CPIOSQ" USING C-RC, SQL-SEL4, SQL-SEL4-LENGTH,
           STAT.
           IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
           PERFORM BVC-RECNUM.
           MOVE 1 TO POS. PERFORM DFC-ABBREV.
           MOVE 2 TO POS.
       DFC-DTNOM.
           CALL "CPIDFC" USING C-RC, POS, DTNOM, DTNOM-LENGTH,
               ASC, SCALE, STAT.
           IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
           MOVE 3 TO POS.
       DFC-DTEXO.
           CALL "CPIDFC" USING C-RC, POS, DTEXO, DTEXO-LENGTH,
               ASC, SCALE, STAT.
           IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
           MOVE 4 TO POS.
       DFC-SITN.
           CALL "CPIDFC" USING C-RC, POS, SITN, SITN-LENGTH,
               ASC, SCALE, STAT.
           IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
       END-SITN.
           PERFORM CALLEXE.
           DISPLAY " ".
           DISPLAY "ABBREV      DTNOM         DTEXO        SITN".
           DISPLAY " ".
       ASKFOUR.
           PERFORM CALLFCH.
           IF STAT = 1403
               GO TO END-FOUR
           ELSE
               DISPLAY ABBREV "    " DTNOM "    " DTEXO "    " SITN
               GO TO ASKFOUR.
       FIM-FOUR.
           DISPLAY "SELECTION ABORTED".
       END-FOUR.
           PERFORM PAUSA THRU END--PAUSA.
      *
      *    SELECT SIX = GIVEN A RECNUM GET ATTACHMENT HISTORICAL
```

```
*
 SELSIX.
     DISPLAY "ENTER RECNUM (12):".
     ACCEPT RECNUM.
     IF RECNUM = " "
         DISPLAY "RECNUM CAN NOT BE NULL"
         DISPLAY "DO YOU WANT TO ABORT THIS SELECTION (Y/N)?"
         ACCEPT Y-N
         IF Y-N = "Y" OR "y"
             GO TO FIM-SIX
         ELSE
             GO TO SELSIX.
     CALL "CPIOSQ" USING C-RC, SQL-SEL6, SQL-SEL6-LENGTH,
     STAT.
     IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
     PERFORM BVC-RECNUM.
     MOVE 1 TO POS. PERFORM DFC-ABBREV.
     MOVE 2 TO POS.
 DFC-DTSATT.
     CALL "CPIDFC" USING C-RC, POS, DTSATT, DTSATT-LENGTH,
         ASC, SCALE, STAT.
     IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
     MOVE 3 TO POS.
 DFC-DTEATT.
     CALL "CPIDFC" USING C-RC, POS, DTEATT, DTEATT-LENGTH,
         ASC, SCALE, STAT.
     IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
     MOVE 4 TO POS.
 DFC-SITA.
     CALL "CPIDFC" USING C-RC, POS, SITA, SITA-LENGTH,
         ASC, SCALE, STAT.
     IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
 END-SITA.
     PERFORM CALLEXE.
     DISPLAY " ".
     DISPLAY "ABBREV       DTSATT       DTEATT      SITA".
     DISPLAY " ".
 ASKSIX.
     PERFORM CALLFCH.
     IF STAT = 1403
         GO TO END-SIX
     ELSE
         DISPLAY ABBREV "    " DTSATT "    " DTEATT "    " SITA
         GO TO ASKSIX.
 FIM-SIX.
     DISPLAY "SELECTION ABORTED".
 END-SIX.
     PERFORM PAUSA THRU END-PAUSA.
*
*    SELECT SEVEN = GIVE A RECNUM GET FLIGHT INFORMATIONS
*
 SELSEVEN.
     DISPLAY "ENTER RECNUM (12):".
     ACCEPT RECNUM.
```

```
    IF RECNUM = " "
        DISPLAY "RECNUM CAN NOT BE NULL"
        DISPLAY "DO YOJ WANT TO ABORT THIS SELECTION (Y/N)?"
        ACCEPT Y-N
        IF Y-N = "Y" OR "y"
            GO TO FIM-SEVEN
        ELSE
            GO TO SELSEVEN.
    CALL "CPIOSQ" USING C-RC, SQL-SEL7, SQL-SEL7-LENGTH,
    STAT.
    IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
    PERFORM BVC-RECNUM.
    MOVE 1 TO POS.
DFC-RANK.
    CALL "CPIDFC" USING C-RC, POS, RANK, RANK-LENGTH, ASC,
        SCALE, STAT.
    IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
    MOVE 2 TO POS.
DFC-YEAREF.
    CALL "CPIDFC" USING C-RC, POS, YEAREF, YEAREF-LENGTH,
        ASC, SCALE, STAT.
    IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
    MOVE 3 TO POS.
DFC-QUAREF.
    CALL "CPIDFC" USING C-RC, POS, QUAREF, QUAREF-LENGTH,
        ASC, SCALE, STAT.
    IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
    MOVE 4 TO POS.
DFN-P1DIU.
    CALL "CPIDFN" USING C-RC, POS, P1DIU, P1DIU-LENGTH,
                                NINETY-NINE, SCALE, STAT.
    IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
    MOVE 5 TO POS.
DFN-P2DIU.
    CALL "CPIDFN" USING C-RC, POS, P2DIU, P2DIU-LENGTH,
                                    NINETY-NINE, SCALE, STAT.
    IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
    MOVE 6 TO POS.
DFN-P1NOC.
    CALL "CPIDFN" USING C-RC, POS, P1NOC, P1NOC-LENGTH,
                                NINETY-NINE, SCALE, STAT.
    IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
    MOVE 7 TO POS.
DFN-P2NOC.
    CALL "CPIDFN" USING C-RC, POS, P2NOC, P2NOC-LENGTH,
                                NINETY-NINE, SCALE, STAT.
    IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
END-P2NOC.
    PERFORM CALLEXE.
    DISPLAY " ".
    DISPLAY "RANK  YEAREF  QUAREF  1PDIU  2PDIU  1PNOC  2PNOC".
    DISPLAY " ".
ASKSEVEN.
    PERFORM CALLFCH.
```

```
     IF STAT = 1403
         GO TO END-SEVEN
     ELSE
         DISPLAY RANK "       " YEAREF "      " QUAREF "     " P1DIU
         "   " P2DIU "  " P1NOC "  " P2NOC
         GO TO ASKSEVEN.
 FIM-SEVEN.
     DISPLAY "SELECTION ABORTED".
 END-SEVEN.
     PERFORM PAUSA THRU END-PAUSA.
*
*    SELECT EIGHT = GIVEN AN UNIT GET RELATED INFORMATIONS
*
 SELEIGHT.
     DISPLAY "ENTER UNIT (6):".
     ACCEPT ABBREV.
     IF ABBREV = " "
         DISPLAY "ABBREV CAN NOT BE NULL"
         DISPLAY "DO YOU WANT TO ABORT THIS SELECTION (Y/N)?"
         ACCEPT Y-N
         IF Y-N = "Y" OR "y"
             GO TO FIM-EIGHT
         ELSE
             GO TO SELEIGHT.
     CALL "CPIOSQ" USING C-RC, SQL-SEL8, SQL-SEL8-LENGTH,
     STAT.
     IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
 BVC-ABBREV.
     CALL "CPIBVC" USING C-RC, ABBREV-N, ABBREV-N-LENGTH,
           ABBREV, ABBREV-LENGTH, ASC, STAT.
     IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
     MOVE 1 TO POS.
 DFC-UNAME.
     CALL "CPIDFC" USING C-RC, POS, UNAME, UNAME-LENGTH,
          ASC, SCALE, STAT.
     IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
     MOVE 2 TO POS.
 DFC-LOCAL.
     CALL "CPIDFC" USING C-RC, POS, LOCAL, LOCAL-LENGTH,
                               ASC, SCALE, STAT.
     IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
     MOVE 4 TO POS.
 DFC-MAJCOM.
     CALL "CPIDFC" USING C-RC, POS, MAJCOM, MAJCOM-LENGTH,
         ASC, SCALE, STAT.
     IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
     MOVE 3 TO POS.
 DFC-REGCOM.
     CALL "CPIDFC" USING C-RC, POS, REGCOM, REGCOM-LENGTH,
         ASC, SCALE, STAT.
     IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
 END-REGCOM.
     PERFORM CALLEXE.
     DISPLAY " ".
```

```
           DISPLAY "  UNIT NAME                              LOCAL      REGCOM
      -    "   MAJCOM".
           DISPLAY " ".
       ASKEIGHT.
           PERFORM CALLFCH.
           IF STAT = 1403
               GO TO END-EIGHT
           ELSE
               DISPLAY UNAME  LOCAL " " REGCOM "       " MAJCOM
               GO TO ASKEIGHT.
       FIM-EIGHT.
           DISPLAY "SELECTION ABORTED".
       END-EIGHT.
           PERFORM PAUSA THRU END-PAUSA.
      *
      *    SELECT NINE = GIVEN AN UNIT GET PERSONS ASSIGNED
      *
       SELNINE.
           DISPLAY "ENTER UNIT (6):".
           ACCEPT ABBREV.
           IF ABBREV = " "
               DISPLAY "ABBREV CAN NOT BE NULL"
               DISPLAY "DO YOU WANT TO ABORT THIS SELECTION (Y/N)?"
               ACCEPT Y-N
               IF Y-N = "Y" OR "y"
                   GO TO FIM-NINE
               ELSE
                   GO TO SELNINE.
           CALL "CPIOSQ" USING C-RC, SQL-SEL9, SQL-SEL9-LENGTH,
           STAT.
           IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
      *
           PERFORM BVC-ABBREV.
           MOVE 1 TO POS. PERFORM DFC-RECNUM.
           MOVE 2 TO POS. PERFORM DFC-TNAME.
           PERFORM CALLEXE.
           DISPLAY " ".
           DISPLAY "RECNUM             TAG NAME".
           DISPLAY " ".
       ASKNINE.
           PERFORM CALLFCH.
           IF STAT = 1403 GO  TO END-NINE.
           DISPLAY RECNUM, "   ", TNAME.
           GO TO ASKNINE.
       FIM-NINE.
           DISPLAY "SELECTION ABORTED".
       END-NINE.
           PERFORM PAUSA THRU END-PAUSA.
      *
      *    SELECT TEN = SELECT UNITS WITH EXCEDENTS (EFFECT > PRED)
      *
       SELTEN.
           CALL "CPIOSQ" USING C-RC, SQL-SEL10, SQL-SEL10-LENGTH,
           STAT.
```

```
     IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
     MOVE 1 TO POS. PERFORM DFC-ABBREV.
     MOVE 2 TO POS. PERFORM DFC-RANK.
     MOVE 3 TO POS.
 DFC-ACTLIST.
     CALL "CPIDFC" USING C-RC, POS, ACTLIST, ACTLIST-LENGTH, ASC,
          SCALE, STAT.
     IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
     MOVE 5 TO POS.
 DFN-PRED.
     CALL "CPIDFN" USING C-RC, POS, PRED, PRED-LENGTH,
         NINETY-NINE, SCALE, STAT.
     IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
     MOVE 4 TO POS.
 DFN-EFFECT.
     CALL "CPIDFN" USING C-RC, POS, EFFECT, EFFECT-LENGTH,
         NINETY-NINE, SCALE, STAT.
     IF STAT NOT = 0 PERFORM ORA-ERR GO TO EXIT-CLOSE.
 END-DFCEFF.
     PERFORM CALLEXE.
     DISPLAY " ".
     DISPLAY "UNIT      RANK  ACTLIST   EFFECTIVE      PREDICT".
     DISPLAY " ".
 ASKTEN.
     PERFORM CALLFCH.
     IF STAT = 1403
         GO TO END-TEN
     ELSE
         DISPLAY ABBREV "      " RANK "    " ACTLIST "   " EFFECT
         "       " PRED
         GO TO ASKTEN.
 FIM-TEN.
     DISPLAY "SELECTION ABORTED".
 END-TEN.
     PERFORM PAUSA THRU END-PAUSA.
*
 HELP.
     DISPLAY "HELP".
 END-HELP.
     EXIT.
 QUIT.
     EXIT PROGRAM.
 END-QUIT.
     EXIT.
*
 EXIT-CLOSE.
     CALL "CPICLS" USING C-RC, STAT.
     IF STAT NOT = 0 PERFORM ORA-ERR.
*
*
 EXIT-LOGOF.
     CALL "CPILOF" USING STAT.
     IF STAT NOT = 0 PERFORM ORA-ERR.
*
```

```
END-OF-PGM.
EXIT-STOP.
    EXIT PROGRAM. STOP-RUN.
*
*    DISPLAY ORACLE ERROR NOTICE
*
ORA-ERR.
    IF STAT NOT = 0 MOVE STAT TO ERR-RC
             MOVE "0" TO ERR-FUNC
    ELSE IF C-RC NOT = 0 MOVE C-RC TO ERR-RC MOVE C-FNC TO
         ERR-FUNC
         ELSE MOVE C-RC TO ERR-RC MOVE C-FNC TO ERR-FUNC.
    MOVE ERR-RC TO ERR-RCX.
    DISPLAY "ORACLE ERROR. CODE IS ", ERR-RCX, ",FUNCTION IS ",
         ERR-FUNC.
    CALL "CPIGEM" USING ERR-RC, MSGBUF, EIGHTY, STAT.
    DISPLAY MSGBUF.
PAUSA.
    DISPLAY " ".
    DISPLAY "HIT ANY KEY TO CONTINUE".
    ACCEPT Y-N.
END-PAUSA.
    EXIT.


******************************************************************
*                                                                *
*     VARIABLES SHARED BY ALL PROGRAMS OF THE PROTOTYPE, KEPT    *
*     IN LIBRARY, CALLD BY "COPY" COMMAND                        *
*                                                                *
******************************************************************
*                                                                *
*                                                                *
*    POS = USED TO SPECIFY THE POSITION OF A FIELD IN A TABLE.
 77  POS                    PIC S9999 COMP VALUE 1.
*    Y-N USED TO ACCEPT ANSWER FROM THE USER.
 77  Y-N                    PIC X VALUE SPACES.
*    NINETY-NINE = DATA TYPE OF THE PGM VARIABLE, 99 FOR NUMERIC.
 77  NINETY-NINE            PIC S9999 VALUE 99 COMP.
*    ASC = DATA TYPE OF THE PGM VARIABLE, ASC FOR CHARACTER.
 77  ASC                    PIC S9999 COMP VALUE 2.
*
*    THE FOLLOWING SET OF SIX VARIABLES, ARE REQUIRED BY ORACLE.
*    THE BASIC COMPOSITION OF A VARIABLE IS ROOT+SUFFIX.
*    THE ROOT IS A REGULAR VARIABLE USED IN THE PERSONNEL
*    DATABASE AND EACH SUFFIX DETERMINES THEIR MEANS.
*
*    SUFFIX          MEANS
*
*    -N-LENGTH       LENGTH OF CHAR STRING SPECIFIED IN SQLVAR
*    -N              SQLVAR, CHAR STRING USED IN SQL COMMAND
*    -LENGTH         LENGTH OF THE PGM VARIABLE.
*    NO SUFFIX       ADDRESS OF THE PGM VARIABLE.
*    -SIZE              ADDRESS OF VARIABLE TO PUT THE FIELD SIZE.
*    -W                 WORK VARIABLE, USED TO SAVE VALUES
*
***************************************************************
*
*    RECNUM = PERSON RECORD NUMBER
*
 77  RECNUM-N-LENGTH      PIC S9999 VALUE 7 COMP.
 77  RECNUM-N             PIC X(7) VALUE ":RECNUM".
 77  RECNUM-LENGTH        PIC S9999 VALUE 12 COMP.
 77  RECNUM-SIZE          PIC S9999 COMP.
 77  RECNUM-W             PIC X(12).
 77  RECNUM               PIC X(12).
*
*    ABBREV = ABBREVIATION OF THE UNIT NAME
*
 77  ABBREV-N-LENGTH      PIC S9999 VALUE 7  COMP.
 77  ABBREV-N             PIC X(7)  VALUE ":ABBREV".
 77  ABBREV-SIZE          PIC S9999.
 77  ABBREV-LENGTH        PIC S9999 VALUE 6  COMP.
 77  ABBREV               PIC X(6)  VALUE " ".
 77  ABBREV-W             PIC X(6).
*
*    ACTLIST = SPECIALTY OF THE PERSON, EX: AV, ENG, MED-DOC
*
 77  ACTLIST-N-LENGTH     PIC S9999 VALUE 8 COMP.
 77  ACTLIST-N            PIC X(8) VALUE ":ACTLIST".
 77  ACTLIST-LENGTH       PIC S9999 VALUE 09 COMP.
 77  ACTLIST-SIZE         PIC S9999 COMP.
 77  ACTLIST-W            PIC X(09).
 77  ACTLIST              PIC  X(09).
*
*    DTACTLIST = DATE OF INCLUSION IN THE ACTLIST
*
 77  DTACTLIST-N-LENGTH   PIC S9999 VALUE 10 COMP.
 77  DTACTLIST-N          PIC X(10) VALUE ":DTACTLIST".
 77  DTACTLIST-LENGTH     PIC S9999 VALUE 09 COMP.
 77  DTACTLIST-SIZE       PIC S9999 COMP.
 77  DTACTLIST-W          PIC X(09).
 77  DTACTLIST            PIC  X(09).
*
*    RANK = MILITARY RANK OF THE PERSON
*
 77  RANK-N-LENGTH        PIC S9999 VALUE 5  COMP.
 77  RANK-N               PIC X(5)  VALUE ":RANK".
 77  RANK-SIZE            PIC S9999 COMP.
 77  RANK-LENGTH          PIC S9999 VALUE 2  COMP.
 77  RANK                 PIC X(2)  VALUE " ".
 77  RANK-W               PIC X(2).
*
*    REAINCL = CODE OF THE SUMMARY OF REASON FOR INCLUSION
*
 77  REAINCL-N-LENGTH     PIC S9999 VALUE 8 COMP.
 77  REAINCL-N            PIC X(8) VALUE ":REAINCL".
 77  REAINCL-LENGTH       PIC S9999 VALUE 08 COMP.
 77  REAINCL-SIZE         PIC S9999 COMP.
 77  REAINCL-W            PIC X(08).
 77  REAINCL              PIC X(8).
*
*    DTINCL = DATE OF INCLUSION IN THE B.A.F.
*
 77  DTINCL-N-LENGTH     PIC S9999 VALUE 7 COMP.
 77  DTINCL-N            PIC X(7) VALUE ":DTINCL".
 77  DTINCL-LENGTH       PIC S9999 VALUE 09 COMP.
 77  DTINCL-SIZE         PIC S9999 COMP.
 77  DTINCL-W            PIC X(09).
 77  DTINCL              PIC X(9).
*
*    SUMINCL = SUMMARY OF THE REASON FOR INCLUSION IN THE B.A.F.
*
 77  SUMINCL-N-LENGTH     PIC S9999 VALUE 8  COMP.
 77  SUMINCL-N            PIC X(8)  VALUE ":SUMINCL".
 77  SUMINCL-SIZE         PIC S9999.
 77  SUMINCL-LENGTH       PIC S9999 VALUE 30 COMP.
 77  SUMINCL              PIC X(30) VALUE " ".
 77  SUMINCL-W            PIC X(30).
*
*    PRED = PREDICTED NUMBER OF PERSONS IN THE UNIT
*
 77  PRED-N-LENGTH        PIC S9999 VALUE 5 COMP.
 77  PRED-N               PIC X(5) VALUE ":PRED".
 77  PRED-LENGTH          PIC S9999 VALUE 03 COMP.
 77  PRED-SIZE            PIC S9999 COMP.
 77  PRED-W               PIC X(05).
 77  PRED                 PIC  S9(5) COMP.
*
*    EFFECT = EFFECTIVE NUMBER OF PERSONS IN THE UNIT
*
 77  EFFECT-N-LENGTH      PIC S9999 VALUE 7  COMP.
 77  EFFECT-N             PIC X(7)  VALUE ":EFFECT".
 77  EFFECT-SIZE          PIC S9999 COMP.
 77  EFFECT-LENGTH        PIC S9999 VALUE 3  COMP.
 77  EFFECT-W             PIC X(5)  VALUE " ".
 77  EFFECT               PIC S9(5) COMP.
*
*    STBIRTH = STATE OF BIRTH
*
 77  STBIRTH-N-LENGTH     PIC S9999 VALUE 8 COMP.
 77  STBIRTH-N            PIC X(8) VALUE ":STBIRTH".
 77  STBIRTH-LENGTH       PIC S9999 VALUE 02 COMP.
 77  STBIRTH-SIZE         PIC S9999 COMP.
 77  STBIRTH-W            PIC X(02).
 77  STBIRTH              PIC X(2).
*
*    TNAME = TAG NAME OF THE PERSON
*
 77  TNAME-N-LENGTH       PIC S9999 VALUE 6  COMP.
 77  TNAME-N              PIC X(6)  VALUE ":TNAME".
 77  TNAME-SIZE          PIC S9999.
 77  TNAME-LENGTH        PIC S9999 VALUE 20 COMP.
 77  TNAME               PIC X(20) VALUE " ".
 77  TNAME-W             PIC X(20).
*
*    UNITH = UNIT THAT ISSUED THE HEALTH CARD
*
 77  UNITH-N-LENGTH      PIC S9999 VALUE 6 COMP.
 77  UNITH-N             PIC X(6) VALUE ":UNITH".
 77  UNITH-LENGTH        PIC S9999 VALUE 06 COMP.
 77  UNITH-SIZE          PIC S9999 COMP.
 77  UNITH-W             PIC X(06).
 77  UNITH               PIC  X(06).
*
*    DTEXPH = EXPIRATION DATE OF THE HEALTH CARD
*
 77  DTEXPH-N-LENGTH     PIC S9999 VALUE 7  COMP.
 77  DTEXPH-N            PIC X(7)  VALUE ":DTEXPH".
 77  DTEXPH-SIZE         PIC S9999.
 77  DTEXPH-LENGTH       PIC S9999 VALUE 9  COMP.
 77  DTEXPH              PIC X(9)  VALUE " ".
 77  DTEXPH-W            PIC X(9).
*
*    CUNIT = CURRENT UNIT WHERE THE PERSON IS ASSIGNED
*
 77  CUNIT-N-LENGTH      PIC S9999 VALUE 6 COMP.
 77  CUNIT-N             PIC X(6) VALUE ":CUNIT".
 77  CUNIT-LENGTH        PIC S9999 VALUE 06 COMP.
 77  CUNIT-SIZE          PIC S9999 COMP.
 77  CUNIT-W             PIC X(06).
 77  CUNIT               PIC  X(06).
*
*    CACTLIST = CURRENT ACTLIST OF THE PERSON
*
 77  CACTLIST-N-LENGTH   PIC S9999 VALUE 9  COMP.
 77  CACTLIST-N          PIC X(9)  VALUE ":CACTLIST".
 77  CACTLIST-SIZE       PIC S9999.
 77  CACTLIST-LENGTH     PIC S9999 VALUE 9  COMP.
 77  CACTLIST            PIC X(9)  VALUE " ".
 77  CACTLIST-W          PIC X(9).
*
*    CRANK = CURRENT RANK OF THE PERSON
*
 77  CRANK-N-LENGTH      PIC S9999 VALUE 6 COMP.
 77  CRANK-N             PIC X(6) VALUE ":CRANK".
 77  CRANK-LENGTH        PIC S9999 VALUE 02 COMP.
 77  CRANK-SIZE          PIC S9999 COMP.
 77  CRANK-W             PIC X(02).
 77  CRANK               PIC  X(02).
*
*    UNAME = UNIT NAME
*
 77  UNAME-N-LENGTH      PIC S9999 VALUE 6 COMP.
 77  UNAME-N             PIC X(6)  VALUE ":UNAME".
 77  UNAME-SIZE           PIC S9999.
 77  UNAME-LENGTH         PIC S9999 VALUE 30 COMP.
 77  UNAME                PIC X(30) VALUE " ".
 77  UNAME-W              PIC X(30).
*
*    LOCAL = LOCALIZATION OF THE UNIT (CITY, COUNTY, ETC.)
*
 77  LOCAL-N-LENGTH       PIC S9999 VALUE 6 COMP.
 77  LOCAL-N              PIC X(6) VALUE ":LOCAL".
 77  LOCAL-LENGTH         PIC S9999 VALUE 15 COMP.
 77  LOCAL-SIZE           PIC S9999 COMP.
 77  LOCAL-W              PIC X(15).
 77  LOCAL                PIC  X(15).
*
*    REGCOM = REGIONAL COMMAND OF SUBORDINATION
*
 77  REGCOM-N-LENGTH      PIC S9999 VALUE 7  COMP.
 77  REGCOM-N             PIC X(7)  VALUE ":REGCOM".
 77  REGCOM-SIZE          PIC S9999.
 77  REGCOM-LENGTH        PIC S9999 VALUE 1  COMP.
 77  REGCOM               PIC X(1)  VALUE " ".
 77  REGCOM-W             PIC X(1).
*
*    MAJCOM = MAJOR COMMAND OF SUBORDINATION
*
 77  MAJCOM-N-LENGTH      PIC S9999 VALUE 7 COMP.
 77  MAJCOM-N             PIC X(7) VALUE ":MAJCOM".
 77  MAJCOM-LENGTH        PIC S9999 VALUE 06 COMP.
 77  MAJCOM-SIZE          PIC S9999 COMP.
 77  MAJCOM-W             PIC X(06).
 77  MAJCOM               PIC  X(06).
*
*    DTMOV = DATE OF MOVING
*
 77  DTMOV-N-LENGTH       PIC S9999 VALUE 6  COMP.
 77  DTMOV-N              PIC X(6)  VALUE ":DTMOV".
 77  DTMOV-SIZE           PIC S9999.
 77  DTMOV-LENGTH         PIC S9999 VALUE 9  COMP.
 77  DTMOV                PIC X(9)  VALUE " ".
 77  DTMOV-W              PIC X(9).
*
*    DTPRES = DATE OF PRESENTATION IN THE ASSIGNED UNIT
*
 77  DTPRES-N-LENGTH      PIC S9999 VALUE 7 COMP.
 77  DTPRES-N             PIC X(7) VALUE ":DTPRES".
 77  DTPRES-LENGTH        PIC S9999 VALUE 09 COMP.
 77  DTPRES-SIZE          PIC S9999 COMP.
 77  DTPRES-W             PIC X(09).
 77  DTPRES               PIC  X(09).
*
*    DTDETACH = DATE OF LEAVING THE UNIT TO THE NEXT ASSIGNMENT
*
 77  DTDETACH-N-LENGTH    PIC S9999 VALUE 9 COMP.
 77  DTDETACH-N           PIC X(9) VALUE ":DTDETACH".
 77  DTDETACH-LENGTH     PIC S9999 VALUE 09 COMP.
 77  DTDETACH-SIZE       PIC S9999 COMP.
 77  DTDETACH-W          PIC X(09).
 77  DTDETACH            PIC  X(09).
*
*    SITU = SITUATION OF THE PERSON IN THE UNIT (EFF, INSTR, ETC.)
*
 77  SITU-N-LENGTH       PIC S9999 VALUE 5 COMP.
 77  SITU-N              PIC X(5) VALUE ":SITU".
 77  SITU-LENGTH         PIC S9999 VALUE 06 COMP.
 77  SITU-SIZE           PIC S9999 COMP.
 77  SITU-W              PIC X(06).
 77  SITU                PIC  X(06).
*
*    DTNOM = DATE OF NOMINATION
*
 77  DTNOM-N-LENGTH      PIC S9999 VALUE 6 COMP.
 77  DTNOM-N             PIC X(6) VALUE ":DTNOM".
 77  DTNOM-LENGTH        PIC S9999 VALUE 09 COMP.
 77  DTNOM-SIZE          PIC S9999 COMP.
 77  DTNOM-W             PIC X(09).
 77  DTNOM               PIC  X(09).
*
*    DTEXO = DATE OF EXONERATION OF THE FUNCTION.
*
 77  DTEXO-N-LENGTH      PIC S9999 VALUE 6 COMP.
 77  DTEXO-N             PIC X(6) VALUE ":DTEXO".
 77  DTEXO-LENGTH        PIC S9999 VALUE 09 COMP.
 77  DTEXO-SIZE          PIC S9999 COMP.
 77  DTEXO-W             PIC X(09).
 77  DTEXO               PIC  X(09).
*
*    SITN = SITUATION OF THE PERSON NOMINATED
*
 77  SITN-N-LENGTH       PIC S9999 VALUE 5 COMP.
 77  SITN-N              PIC X(5) VALUE ":SITN".
 77  SITN-LENGTH         PIC S9999 VALUE 06 COMP.
 77  SITN-SIZE           PIC S9999 COMP.
 77  SITN-W              PIC X(06).
 77  SITN                PIC  X(06).
*
*    DTDES = DATE OF DESIGNATION
*
 77  DTDES-N-LENGTH      PIC S9999 VALUE 6 COMP.
 77  DTDES-N             PIC X(6) VALUE ":DTDES".
 77  DTDES-LENGTH        PIC S9999 VALUE 09 COMP.
 77  DTDES-SIZE          PIC S9999 COMP.
 77  DTDES-W             PIC X(09).
 77  DTDES               PIC  X(09).
*
*    DTWAIVER = DATE OF WAIVER FROM THE DESIGNATION
*
 77  DTWAIVER-N-LENGTH   PIC S9999 VALUE 9 COMP.
 77  DTWAIVER-N          PIC X(9) VALUE ":DTWAIVER".
 77  DTWAIVER-LENGTH      PIC S9999 VALUE 09 COMP.
 77  DTWAIVER-SIZE        PIC S9999 COMP.
 77  DTWAIVER-W           PIC X(09).
 77  DTWAIVER             PIC  X(09).
*
*    SITD = SITUATION OF THE PERSON DESIGNATION
*
 77  SITD-N-LENGTH        PIC S9999 VALUE 5 COMP.
 77  SITD-N               PIC X(5) VALUE ":SITD".
 77  SITD-LENGTH          PIC S9999 VALUE 06 COMP.
 77  SITD-SIZE            PIC S9999 COMP.
 77  SITD-W               PIC X(06).
 77  SITD                 PIC  X(06).
*
*    DTSATT = DATE OF START THE ATTACHMENT
*
 77  DTSATT-N-LENGTH      PIC S9999 VALUE 7 COMP.
 77  DTSATT-N             PIC X(7) VALUE ":DTSATT".
 77  DTSATT-LENGTH        PIC S9999 VALUE 09 COMP.
 77  DTSATT-SIZE          PIC S9999 COMP.
 77  DTSATT-W             PIC X(09).
 77  DTSATT               PIC  X(09).
*
*    DTEATT = DATE OF END THE ATTACHMENT
*
 77  DTEATT-N-LENGTH      PIC S9999 VALUE 7 COMP.
 77  DTEATT-N             PIC X(7) VALUE ":DTEATT".
 77  DTEATT-LENGTH        PIC S9999 VALUE 09 COMP.
 77  DTEATT-SIZE          PIC S9999 COMP.
 77  DTEATT-W             PIC X(09).
 77  DTEATT               PIC  X(09).
*
*    SITA = SITUATION OF THE PERSON ATTACHMENT
*
 77  SITA-N-LENGTH         PIC S9999 VALUE 5 COMP.
 77  SITA-N                PIC X(5) VALUE ":SITA".
 77  SITA-LENGTH           PIC S9999 VALUE 06 COMP.
 77  SITA-SIZE             PIC S9999 COMP.
 77  SITA-W                PIC X(06).
 77  SITA                  PIC  X(06).
*
*    YEAREF = YEAR OF REFERENCE FOR THE FLIGHT
*
 77  YEAREF-N-LENGTH      PIC S9999 VALUE 7 COMP.
 77  YEAREF-N             PIC X(7) VALUE ":YEAREF".
 77  YEAREF-LENGTH        PIC S9999 VALUE 02 COMP.
 77  YEAREF-SIZE          PIC S9999 COMP.
 77  YEAREF-W             PIC X(02).
 77  YEAREF               PIC  X(02).
*
*    QUAREF = QUATER OF REFERENCE FOR THE FLIGHT
*
 77  QUAREF-N-LENGTH      PIC S9999 VALUE 7 COMP.
 77  QUAREF-N             PIC X(7) VALUE ":QUAREF".
 77  QUAREF-LENGTH        PIC S9999 VALUE 01 COMP.
 77  QUAREF-SIZE          PIC S9999 COMP.
 77  QUAREF-W             PIC X(01).
 77  QUAREF               PIC  X(01).
*
*    P1DIU = HOURS FLEW AS FIRST PILOT DIURN
*
 77  P1DIU-N-LENGTH       PIC S9999 VALUE 6 COMP.
 77  P1DIU-N              PIC X(6) VALUE ":P1DIU".
 77  P1DIU-LENGTH         PIC S9999 VALUE 06 COMP.
 77  P1DIU-SIZE           PIC S9999 COMP.
 77  P1DIU-W              PIC X(05).
 77  P1DIU                PIC S9(4)V9 COMP.
*
*    P2DIU = HOURS FLEW AS SECOND PILOT DIURN
*
 77  P2DIU-N-LENGTH       PIC S9999 VALUE 6 COMP.
 77  P2DIU-N              PIC X(6) VALUE ":P2DIU".
 77  P2DIU-LENGTH         PIC S9999 VALUE 03 COMP.
 77  P2DIU-SIZE           PIC S9999 COMP.
 77  P2DIU-W              PIC X(05).
 77  P2DIU                PIC S9(4)V9  COMP.
*
*    P1NOC = HOURS FLEW AS FIRST PILOT NOCTURN
*
 77  P1NOC-N-LENGTH       PIC S9999 VALUE 6 COMP.
 77  P1NOC-N              PIC X(6) VALUE ":P1NOC".
 77  P1NOC-LENGTH         PIC S9999 VALUE 03 COMP.
 77  P1NOC-SIZE           PIC S9999 COMP.
 77  P1NOC-W              PIC X(05).
 77  P1NOC                PIC S9(04)V9 COMP.
*
*    P2NOC = HOURS FLEW AS SECOND PILOT NOCTURN
*
 77  P2NOC-N-LENGTH       PIC S9999 VALUE 6 COMP.
 77  P2NOC-N              PIC X(6) VALUE ":P2NOC".
 77  P2NOC-LENGTH         PIC S9999 VALUE 03 COMP.
 77  P2NOC-SIZE           PIC S9999 COMP.
 77  P2NOC-W              PIC X(05).
 77  P2NOC                PIC S9(04)V9 COMP.
*
*    DTPRO = DATE OF PROMOTION
*
 77  DTPRO-N-LENGTH       PIC S9999 VALUE 6 COMP.
 77  DTPRO-N              PIC X(6) VALUE ":DTPRO".
 77  DTPRO-LENGTH         PIC S9999 VALUE 09 COMP.
 77  DTPRO-SIZE           PIC S9999 COMP.
 77  DTPRO-W              PIC X(09).
 77  DTPRO                PIC  X(09).
*
*    CRITPRO = CRITERION OF PROMOTION
*
 77  CRITPRO-N-LENGTH     PIC S9999 VALUE 8 COMP.
 77  CRITPRO-N            PIC X(8) VALUE ":CRITPRO".
 77  CRITPRO-LENGTH       PIC S9999 VALUE 01 COMP.
 77  CRITPRO-SIZE         PIC S9999 COMP.
 77  CRITPRO-W            PIC X(01).
 77  CRITPRO              PIC  X(01).
*
*    VARIABLES USED TO TREAT ERROR CONDITIONS
*
 77  ERR-RC               PIC S9999 COMP.
 77  ERR-RCX              PIC S9999 SIGN LEADING SEPARATE DISPLAY.
 77  ERR-FUNC             PIC S9999 SIGN LEADING SEPARATE DISPLAY.
 77  C-FNC                PIC S9999 COMP.
*
*    VARIABLE USED TO PASS PARAMETERS ON ORACLE
*
 77  SCALE                PIC  S9999 VALUE 0 COMP.
 77  STAT                 PIC S9999 VALUE 0    COMP.
*
*    C-RC = CURSOR USED ON ORACLE
*
 77  C-RC                 PIC S9999 COMP.
*
*    VARIABLES USED ON ORACLE
*
 77  EIGHTY               PIC S9999 VALUE 80 COMP.
 77  MSGBUF               PIC X(80).
 77  TWO                  PIC S9999 VALUE 2 COMP.
 77  ONE                  PIC S9999 VALUE 1 COMP.
 77  SIX                  PIC S9999 VALUE 6 COMP.
 77  CURSOR-SIZE          PIC S9999 VALUE 5000 COMP.
*
*    RESP = AREA USED TO RECEIVE ANSWER FROM THE USER
*    RESP-OK = VALUES ACCEPTED.
*
 77  RESP             PIC XX JUST RIGHT.
 88  RESP-OK          VALUES ARE "1" "2" "3" "4" "5" "6" "7" "8"
                      "9" "10" "11" "12" "13".
*
*****************  END OF VARIABLES  ***************************
```

Bibliography

1. Abel, P., Cobol Programming A Structured Approach, Reston, Virginia: Prentice-Hall, 1980.

2. Astrahan, M.M. et all, "System R: Relational Approach to Database Management.", ACM TODS 1, No. 2, June 1976.

3. Atre, S., Data base: Structured Techniques for Design, Performance, and Management. New York: Wiley-Interscience Publication, 1980.

4. Baek, Maj Kwang Soo, Database Design for Personnel Management in Republic of Korea Army, Naval Postgraduate School, Monterey, CA, June 1984.

5. Brown, R.R. and T.L. Ramey, "The Concept and Practice of ERA Information Modeling", Entity Relationship Approach to System Analysis and Design P.P. Chen (ed.), North-Holland, June 1980.

6. Buykoner, Maj Bora and Cap Yucel Ozin, Design and Implementation of a Personnel Database, MS Thesis. Naval Postgraduate School, Monterey, CA, June 1985.

7. Chen, P.P., Entity Relationship Approach to System Analysis and Design, North-Holland Publishing Co., June 1980.

8. ---------. "The Entity_Relationship Model Toward a Unified View of Data", ACM Transaction on Database System, Vol. 1, No. 1, June 1976.

9. ---------. "Entity_Relationship Approach to Logical Database Design", Q.E.D. Information Science Inc., Wellesley, MA, 1977.

10. ----------. "A Preliminary Framework for Entity_Relationship Models", Entity Relationship Approach to Systems Analysis and Design, P.P. Chen (ed.), North-Holland, 1983.

11. Codd, E.F., "Further Normalization of the Data Base Relational Model", Data Base Systems, Courant Computer Science Symposia Series, Vol.6. Englewood Cliffs, N.J., Prentice-Hall, 1972.

12. ----------. "Recent Investigations into Relational Data Base Systems", Proc. IFIP Congress, 1974.

13. Costa, Sergio Leal da, "Logic Project for Military Personnel Headquarters", 1983.

14. Cunha, Maj Adilson M. da, A Database Design for the Brazilian Air Force Flying Unit Operational Control System, MS Thesis GCS/EE/84D-7. School of Engineering, Air Force Institute of Technology, Wright Patterson Air Force Base, OH, December 1984.

15. Date, C. J., An Introduction to Database Systems (fourth edition). Readings, Massachusetts: Addilson-Wesley Publishing Company. 1986.

16. Fagin, R., "Multivalued Dependencies and a New Normal Form for Relational Databases", ACM TODS 2, No. 3, September, 1972.

17. ----------. "Normal Forms and Relational Database Operators.", Proc. 1979 ACM SIGMOD International Conference on Management of Data.

18. Fairley, R, Software Engineering Concepts. New York, Mc-Graw-Hill, 1984.

19. Gane, C. and Trish Sarson, Structured Systems Analysis: Tools and Techniques. Englewood Cliffs, NJ: Prentice-Hall Inc., 1979.

20. Hammer, D. and D. McLeod, "The Semantic Model: A Modeling Mechanism for Database Applications", ACM SIGMOD-Proceedings, International Conference on Management of Data, Austin, Texas, May 1978.

21. Jones, Cap Peter L., Design and Implementation of a Personnel Database Management System, MS Thesis, Naval Postgraduate School, Monterey, CA, June 1982.

22. Korth, H.F, and A. Silberschatz, Database Systems Concepts., McGraw-Hill Book Company, 1986.

23. Maier, David, The Theory of Relational Databases, Rockville: Computer Science Press, 1983.

24. Martin, James et all, The James Martin Report on High-Productivity Languages, High Productivity Software,Inc. Marblehead, MA, 1986.

25. McLeod, D., A Semantic Data Model and Its Associated Structured User Interface, Ph.D. Dissertation, MIT, Cambridge, MA, 1975.

26. Moedjiono, Cap, The Preliminary Personnel Database Design for the Indonesian Army, MS Thesis, Naval Postgraduate School, Monterey, CA, June 1982.

27. Roth, 2LT Mark A., The Design and Implementation of a Pedagogical Relational Database System, MS Thesis GSC/EE/79-14 School of Engineering, Air Force Institute of Technology, Wright Patterson Air Force Base, OH December 1979.

28. Sari, Lt Osman, Design and Implementation of Inventory Database, MS Thesis, Naval Postgraduate School, Monterey, CA, June 1985.

29. Silva, Victorio B. da, " The Informatic and the Development of its Applications in the Brazilian Air Force", Brazilian Air Staff Command School Project. December 1982.

30. Sprangue, Ralph H. Jr. and Eric D. Carlson, Building Effective Decision Support System. Englewood Cliffs, NJ: 1982.

31. Teichroew, d., et al. "Applications of the Entity_Relationship Approach", Entity Relationship Approach to Information Modeling and Analysis, P.P. Chen (ed.), North-Holland, 1983.

32. Theorey, Toby J. and James P. Fry, Design of Database Structures. Englewood Cliffs, NJ: Prentice-Hall Inc., 1982.

33. Ullman, Jeffrey D. Principals of Database Systems, Rockville: Computer Science Press, 1982.

34. Wiederhold, Gio and El-Masri, "A Structural Model for Database Systems", STAN-CS-79-722, Computer System Department, Stanford University, Feb. 1979.

VITA

Major Wagner Mussato was born on 29 October 1948 in Sao Paulo, Sao Paulo, Brazil. He graduated in 1970 with a Bachelor of Science (B.S.) degree from Academia da Forca Aerea Brasileira (Brazilian Air Force Academy) when he also received his pilot's wings. In 1973 he graduated with a B.S. degree from the Sociedade Unificada de Ensino Superior Augusto Motta (United Society of College Study Augusto Motta).

He has been working with computers since 1973, due to his assignment to Centro de Computacao da Aeronautica (Brazilian Air Force Computational Center). He worked as system analyst in the development of several systems for the Brazilian Air Force. He was Head of the Software Development Division in that Computational Center, until entering the Master's degree course at the Air Force Institute of Technology School of Engineering, in June 1985.

Permanent Address: Rua Aureliano Pimentel 68
Rio de Janeiro, RJ 21931
BRASIL

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION
Unclassified

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
Aproved for public release,
distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
AFIT/GCS/ENG/87J-1

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
School of Engineering

6b. OFFICE SYMBOL *(If applicable)*
AFIT/ENG

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS *(City, State, and ZIP Code)*
Air Force Institute of Technology (AU)
Wright-Patterson AFB, Ohio 45433-6583

7b. ADDRESS *(City, State, and ZIP Code)*

8a. NAME OF FUNDING / SPONSORING ORGANIZATION
Brazilian Air Force

8b. OFFICE SYMBOL *(If applicable)*
BAF

9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS *(City, State, and ZIP Code)*

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO.

11. TITLE *(Include Security Classification)*
A DATABASE DESIGN FOR THE BRAZILIAN AIR FORCE
MILITARY PERSONNEL CONTROL SYSTEM

Approved for public release: IAW AFR 190-1.
Lynn Wolaver 4 Jun 87
LYNN E WOLAVER
Dean for Research and Professional Development
Air Force Institute of Technology (AFC)
Wright-Patterson AFB OH 45433

12. PERSONAL AUTHOR(S)
Wagner Mussato, Major, BAF

13a. TYPE OF REPORT
MS Thesis

13b. TIME COVERED
FROM ________ TO ______

14. DATE OF REPORT *(Year, Month, Day)*
1987 June

15. PAGE COUNT
181

16 SUPPLEMENTARY NOTATION

17 COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Relational Database
Decision Support System
Entity Relationship Model
Normalization Theory
Query Language

19 ABSTRACT *(Continue on reverse if necessary and identify by block number)*

This thesis addresses a database design with partial implementation for the Brazilian Air Force Military Personnel Control System. After defining the problem and specifying requirements, the conceptual design was performed using Entity Relationship Model. After defining the Entities and Relationships, the Normalization Theory was used to ensure that all relations met the constraints of the Fourth Normal Form (4NF).

During the implementation phase, a prototype was implemented using Oracle DBMS, with SQL as a query language and Cobol as a host language. The Decision to use this environment for the implementation was made because SQL and Cobol are the languages used in the Brazilian Air Force.

Finally, recommendations were proposed for future research in this area, along with an optimal environment for a database user, combining mainframe and personal computers.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED □ SAME AS RPT □ DTIC USERS

21 ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL
Mark Roth, Captain, USAF

22b TELEPHONE *(Include Area Code)*
(513) 255-3576

22c OFFICE SYMBOL
AFIT/ENG

Previous editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

END
8-87
DTIC